- A -

A META-PROTOCOL ARCHITECTURE FOR CONNECTING COMPUTER NETWORKS

J. Brad Lindsey

Department of Computer Science

Brigham Young University

M.S. Degree, December 1987

ABSTRACT

Accompanying the proliferation of computer networks has been a movement to connect them into cooperating internets. However, when attempting to do so, the different protocols used to satisfy these once isolated networks are found to incompatible. Due to its reliable nature, the transport layer from ISO's OSI Reference Model is chosen as the point of attachment for subnets and internet gateways. In this role, it is expected to supply traditional transport and inherited services. A meta-protocol architecture is proposed to relay these services from one subnet to the next, until internet messages arrive at their destination. The architecture is based upon each subnet providing two conversion routines -- one from the subnet protocol to the meta-protocol, the other, back to its own protocol. A simulated internet, demonstrating the capabilities of the meta-protocol approach, is described.

Keywords: Thesis, Information exchange. (KR)

by

Joel Brad Lindsey, Capt

U.S. Air Force

79 pages in completed thesis

A META-PROTOCOL ARCHITECTURE FOR CONNECTING COMPUTER NETWORKS

A Thesis

Presented to the

Department of Computer Science

Brigham Young University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

J. Brad Lindsey

December 1987

This thesis by J. Brad Lindsey is accepted in its present form by the Department of Computer Science of Brigham Young University as satisfying the thesis requirement for the degree of Master of Science.

Robert N. Linebarger,
Committee Chairman

Evan L. Ivie, Committee Member

4 Dec. 87
Date

Gordon E. Stokes,
Graduate Coordinator

ACKNOWLEDGEMENTS

I express appreciation, first and foremost, to my sweetheart wife, Wendy, and my two sons, Nicholas and Jared, who have endured much more than I during the preparation of this thesis. For Dr. Robert Linebarger, I have the utmost respect and express my deepest gratitude for supplying not only technical expertise and encouragement, but, more importantly, an example of a true professional that I will retain for the rest of my life -- Thank You!

TABLE OF CONTENTS

Acceptance page ii

Acknowledgements iii

List of Figures and Tables vi

Chapter 1 Introduction 1
1.1 Computer Networks - What 2
1.2 Computer Networks - Why 3
1.2.1 Economic Benefits 3
1.2.2 Technical Benefits 5
1.3 Internets - What 6
1.4 Internets - Why 8
1.5 Computer Networks - How 9
1.6 Internets - How 11
1.6.1 Using Existing Protocol Suites 11
1.6.2 Using an Additional Layer 12

Chapter 2 Internetwork Layer Determination 13
2.1 Layer Analysis 13
2.2 Internetwork Layer Justification 15
2.2.1 Independence From Network-Unique Features 15
2.2.2 Transmission Optimization 15
2.2.3 Information Enveloping 16
2.2.4 Own Addressing Scheme 16
2.2.5 End-to-End Reliability/Correctness 18
2.3 Summary 18

Chapter 3 Transport Protocol-based Internet Services 20
3.1 Traditional Transport Services
3.1.1 Connection Management Mechanisms 23
3.1.2 Transparent Message Delivery Mechanisms 25
3.1.2.1 Positive Acknowledgement with Retransmission (PAR) 25
3.1.2.2 Flow Control Windows 26
3.1.2.3 Duplicate and Out-of-order Detection 27
3.1.2.4 Checksum 27
3.2 Traditional Network Services 28
3.2.1 Name-Address Resolution 28
3.2.2 Message Fragmentation/Reassembly 29
3.3 Heterogeneous Internet Services 29
3.3.1 Sequence Preservation 29
3.3.2 Option Preservation 31

Chapter 4 Internet Architecture 34
4.1 Logical View of the Architecture 35
4.2 Name-to-Address Resolution Sub-layer 38
4.2.1 Flat Address Space 39
4.2.2 Hierarchical Address Space 39
4.2.3 Meta-Protocol Name-to-Address Resolution 40
4.3 Optional Parameter Preservation Sub-layer 47
4.4 Traditional Services Sub-layer 49
4.5 Fragmentation Sub-layer 50
4.5.1 Internet Fragmentation 50
4.5.2 Intranet Fragmentation 50
4.5.2.1 TCP to TP 4 Subnets 51
4.5.2.2 TP 4 to TCP Subnets 52
4.6 Sequence Number Preservation Sub-layer 52
4.7 Summary 53

Chapter 5 Alternative Internetworking Strategies 55
5.1 Global Internetwork Standard 55
5.2 Internet Protocol 56
5.3 Pure Protocol Translator 57

Chapter 6 Meta-Protocol Simulation 59
6.1 Simulated Internet Topology 62
6.2 Message Exchange Scenarios 62
6.2.1 Same Subnet 63
6.2.2 Adjacent, Homogeneous Subnets 63
6.2.3 Adjacent, Heterogeneous Subnets 64
6.2.4 Non-adjacent, Homogeneous Subnets 64
6.2.5 Non-adjacent, Heterogeneous Subnets 66
6.3 Summary 66

Chapter 7 Summary and Conclusions 67
7.1 Summary 67
7.2 Conclusions 68
7.3 Areas for Further Research 69
7.3.1 Name-to-Address Resolution 69
7.3.2 Option Preservation 69
7.3.3 Additional Transport Protocols 70

Endnotes 71

Cited Works 76

LIST OF FIGURES AND TABLES

Chapter 1
Figure 1-1. BACKBONE INTERNET MODEL 7
Figure 1-2. CATENET INTERNET MODEL 8
Figure 1-3. COMMUNICATION PROTOCOL SUITES 9
Figure 1-4. OSI REFERENCE MODEL 10

Chapter 2
Figure 2-1. DUAL NATURE OF TRANSPORT PROTOCOLS 14
Figure 2-2. ENTITIES, CEPs, AND SAPs 17
Figure 2-3. SIGNAL-DATA-INFORMATION TRANSFORMATION 19

Chapter 3
Figure 3-1a. TCP-BASED ENVELOPES 21
Figure 3-1b. TP 4-BASED ENVELOPES 22
Figure 3-2. CONNECTION ESTABLISHMENT 24
Figure 3-3. CONNECTION RELEASE 24
Figure 3-4. SEQUENCE NUMBER EXCHANGE 30
Table 3-1. TCP-BASED OPTIONS 32
Table 3-2. TP 4-BASED OPTIONS 32

Chapter 4
Figure 4-1. META-PROTOCOL INTERNET 35
Figure 4-2. FUNCTIONS OF META-PROTOCOL GATEWAY HALVES 36
Figure 4-3. META-PROTOCOL SUITE 38
Figure 4-4. TCP and TP 4 ADDRESSES 41
Figure 4-5. HOST NAME LOOKUP PROCEDURES 43
Figure 4-6. GATEWAY INTERNET VALUE LOOKUP PROCEDURES 45
Figure 4-7. TRANSPORT PROTOCOL SUB-LAYERS 46
Figure 4-8. OPTION VECTOR MULTIPLICATION 48
Table 4-1. ENVELOPE FIELDS USED FOR FRAGMENTATION 51

Chapter 5
Figure 5-1. COMMON INTERNET PROTOCOL 56
Figure 5-2. PURE PROTOCOL TRANSLATOR 57

Chapter 6
Figure 6-1. INCREMENTAL MESSAGE REPRESENTATION 61
Figure 6-2. SIMULATION TOPOLOGY 63
Figure 6-3. OPTION EXCHANGE SCENARIOS 65

CHAPTER 1

Introduction

Since 1978 the Department of Defense (DOD) has recognized the Transmission Control Protocol (TCP) as its official transport protocol standard for computer networks. In 1984 the International Organization for Standardization's (ISO) Class 4 Transport Protocol (TP 4), functionally equivalent to TCP in many ways, obtained Draft International Standard status and is expected to become the preferred transport protocol for future networks. [1] The acceptance of TP 4 is evidenced by the fact that even the DOD has committed to eventually using it in favor of TCP. [2]

In recent years there has been great emphasis placed on interconnecting autonomous computer networks into integrated "networks of networks", or internets, with the DOD playing an important role in these efforts. For the remainder of this thesis the term "subnet" will refer to a network subscribing to the services of an internet while a "network" will denote a stand-alone computer network. The Defense Data Network (DDN) is an operational military internet linking many TCP-based subnets together. As the DOD begins its migration to ISO's TP 4 protocol, they will be faced with a serious internetworking dilemma-making existing TCP subnets and newly created TP 4 subnets interoperable. [3] This situation will most likely be of concern for many years since there has been a tremendous amount of money invested in TCP-based systems and the DOD will want to utilize them to their full potential.

This thesis proposes a protocol conversion architecture for overcoming the problem soon to face the DOD and which may also affect other organizations. Before addressing transport protocol specifics and the proposed architecture itself, the characteristics of networks and internets will be reviewed. Chapter two will

discuss the different layers at which internetworking may be performed within a protocol suite, then provide justification for using the transport layer for this thesis. Chapter three will describe the services traditionally associated with transport protocols along with those inherited due to its internetworking role. Although TCP and TP 4 are **functionally** equivalent in many ways, such a discussion will uncover obvious **structural** inconsistencies between them. Chapter four will characterize the details of the conversion architecture, showing how it overcomes the inconsistencies of chapter three. Chapter five will discuss alternative internetworking approaches and why the conversion architecture was chosen. Chapter six describes a primitive implementation of the architecture simulating the exchange of information between TCP and TP 4 subnets. Using the results of the simulation, chapter seven will draw general conclusions as to the effectiveness of the architecture and will make recommendations for further areas of study.

1.1 Computer Networks - What

To establish a frame of reference, let us define a network as, "..a set of autonomous, independent computer systems, interconnected so as to permit interactive resource sharing between ary pair of systems." [4] There are two primary methods for providing this sharing between systems -- connectionless and connection-oriented exchange. Connection-oriented techniques maintain state information regarding explicit connections between communicating parties. Resources are allocated at connection establishment time and used for all subsequent message exchange. Connectionless communication, on the other hand, have no concept of connections. Every message to be transferred contains addressing and other information needed to get it from source to destination. Using this information, connectionless techniques treat each message

independently. [5] For this thesis, an assumption is made that the underlying communication sub-system (CSS - more clearly defined in Chapter two) is of the connectionless variety, while the functions performed above the CSS result in connection-oriented information exchange. Regardless of the exchange mechanism, the motivating forces behind computer networks are both economic and technical.

1.2 Computer Networks - Why

1.2.1 Economic Benefits

The use of microprocessor-based computer systems has changed the way information is collected and used in many organizations. The performance of microcomputers now rivals that of previously used mini and mainframe computers. This coupled with the tremendous price differential (perhaps a single mainframe costing a thousand times more than a microcomputer) leads to the conclusion that several microcomputers networked together could provide substantial cost/performance improvements.

In fact, if we consider computing to be just another marketable commodity, it may be possible for an organization to completely do away with its own resources by becoming a subscriber to a commercial network providing computer services. [6], [7] Such an approach would not only remove the cost of the actual equipment, but also the cost of equipment operators and maintenance staff. The supplier of networking technology would also benefit by recovering its investment more quickly through service charges to its subscribers. Another economic benefit of computer networks is characterized by the current trend of increasing communication costs as compared to computing costs. [8]

When relying upon mini and mainframe computation it was infeasible for organizations to place expensive pieces of hardware at each of its sites. The normal mode of operation for gathering information was to relay a collection of data from each site to some central computing center where the detailed analytical functions would be performed. The results would then be sent back to each individual site (many communication transactions per computation). With the arrival of the microprocessor it became more efficient to analyze data at each location and only send periodic updates to the central office for administrative reasons (many computations per communication transaction). Although the absolute cost of communications has not changed dramatically, its cost relative to microprocessor-based computational resources has increased substantially and must be held to a minimum in today's world. A final economic advantage of computer networks has nothing to do with costs of the various systems, but rather the savings realized in productivity.

With the evolution of the industrial world into one of increased cooperation among organizational departments has come the need to share information. One possibility for providing this exchange comes through each department physically transporting material into the central office where corporate management would make its ultimate decisions. A more promising alternative became available when computer networking came of age. The ability to electronically exchange ideas among network subscribers in a matter of seconds rather than hours/days has seen significant increases in decision-making effectiveness. Subsequently, organizations have realized significant economic gains.

The reasons discussed thus far provide solid managerial support for implementing computer networks; the technical benefits are just as convincing.

1.2.2 Technical Benefits

The ultimate benefit obtained from any computer network is that of creating a communication path between subscribers who were previously not connected, thereby providing resources beyond those available from a single computer system. [9] A connectionless CSS technique known as "load-splitting" makes this process very resilient to changes in the condition of network systems. Load-splitting techniques allow separate pieces of an overall message to traverse more than one path within the network before reaching the destination. In the process of determining paths, load-splitting techniques attempt to balance the traffic such that no one path becomes overloaded. If one or more systems in the network become inoperative, the load-splitting CSS is able to divert traffic around the failed site by using an alternative path. [10] While load-splitting benefits subscribers of a network by getting all information from source to destination, additional benefits are realized through the customizing of particular systems to handle only certain types of information.

As discussed previously, one of the economic benefits of computer networking comes from the fact that analytical functions may be distributed across remote locations with the results being available to other subscribers of the network. A natural extension of this leads to a very efficient method of utilizing each computer on a network -- task specialization. For example, if a given subscriber, X, by reason of special software or hardware, is particularly adept at matrix multiplication, one may expect that other subscribers in the network will exploit this capability by multiplying their matrices at X in preference to doing so locally. [11] Local area networks are particularly suited for this type of operation, and special computer systems known as "servers" are designed to handle functions of a given type. For instance, there are file servers

responsible for opening/accessing/closing/deleting files needed by more than one subscriber of the network. Job servers take care of requests to compile computation-extensive programs [12] and archive servers allow for a centralized backup facility supporting any member of the network desiring its services. [13] This centralization of function provides for more efficient information management and reduces the load on other network subscribers by only requiring special programming to be present at server systems. [14] One final comment about using servers on a network that has implications for the architecture of this thesis -- when requests to the server are sent in a standard format, a simple conversion from the local operating system format to the server format allows a heterogeneous mix of remote sites to access the server. [15] The last advantage of networking, from a technical perspective, is a by-product of task specialization.

Since each subscriber is not expected to perform all functions, a simpler software design may be used. In a typical mainframe computer, processing time must be scheduled between different application users (database queries, compilations, graphics, etc.). The scheduling software itself demands a portion of the processing time. In a networking environment, systems are responsible for one task (if taken to the extreme), thereby eliminating the overhead of scheduling software.

Thus far we have been concerned with the benefits derived from connecting independent computer systems. If computer networks are connected to other networks the same benefits exist, only on a larger scale.

1.3 Internets - What

As might be expected, networks exhibit different performance characteristics. As users of independent networks became aware of these capabilities the concept of connecting them together, thereby allowing subscribers on each subnet access

to the functions found in other subnets, became attractive. At the topological level there are two models for creating an internet. Using the first, illustrated in figure 1-1, [16] each subscriber subnet may be located on the periphery of a collection of inner connections, much the way spokes radiate out from the hub of a wheel. The hub connections form a "backbone" that provides the connectivity between internet subscribers. The second model is illustrated in figure 1-2 [17] and is analogous to independent packet-switched networks with the communication paths being entire subnets rather than physical transmission media. [18] In both cases special pieces of equipment known as **gateways** serve as switches and possess varying amounts of intelligence in order to forward traffic from subnet to subnet. The latter model, known as a **"catenet"** [19], shall be used for this thesis.

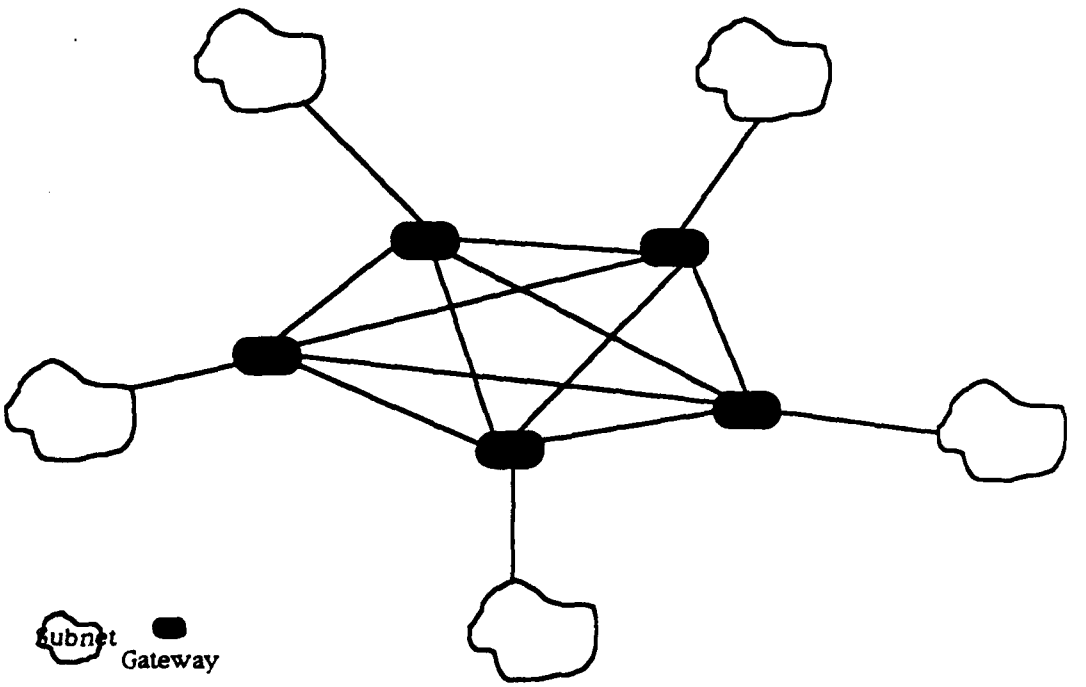


Figure 1-1. BACKBONE INTERNET MODEL

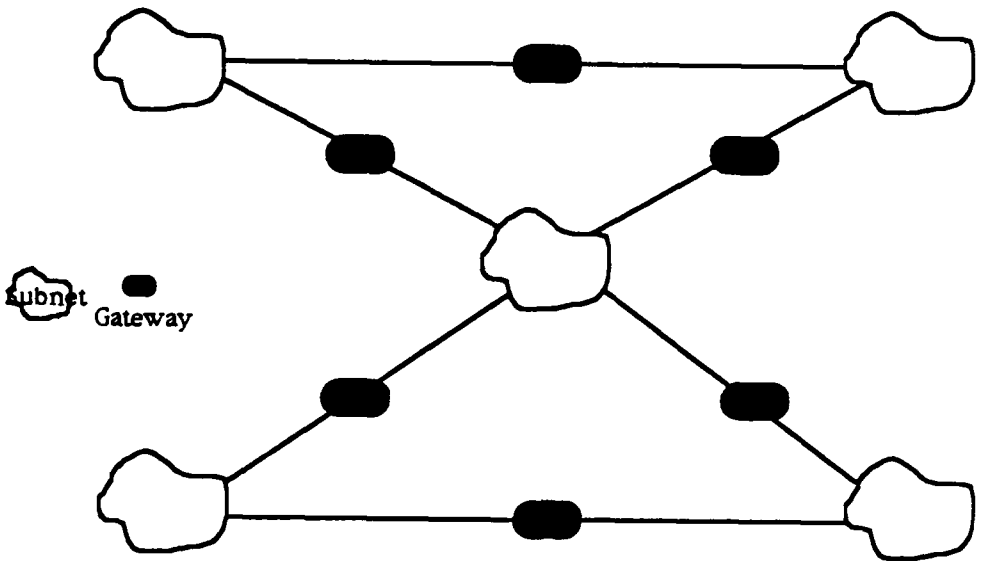


Figure 1-2. CATENET INTERNET MODEL

1.4 Internets - Why

In general, the same benefits discussed previously with regard to individual computer networks apply when considering an internet, but internetworking also overcomes some limitations inherent to separate networks. For instance, local area network performance suffers as the number of attached stations increases. By connecting several smaller subnets, improved performance is realized, especially if subnets are created such that intranet traffic exceeds internet traffic. [20] Individual networks are designed for a particular type of user [21], [22] -- packet-switched networks provide connectivity to a great number of users spread over a wide geographic area. Local area networks connect fewer users with limited coverage, but are able to use bandwidths significantly greater than packet-switched networks. Circuit-switched networks provide quick delivery of data after connection establishment, but suffer from supporting only one conversation/host at a time (as opposed to packet-switch where concurrent conversations are possible). Finally, point-to-point connections might be desired when the two points have an extremely large amount of traffic passing between

them. Internetworking makes it possible to connect these heterogeneous user groups, while attempting to keep the autonomous nature of each group's subnet in tact.

Up to this point we have not said anything about how networks/internets are created, only that they are desirable. The architectural model used in many of today's computer network designs, and subsequent internet designs, will be described next.

1.5 Computer Networks - How

To reduce the amount of time spent in the design phase of communication protocol suites (see figure 1-3), a common architectural model is necessary. Such a model would not stifle the creativity of software engineers, but would enhance

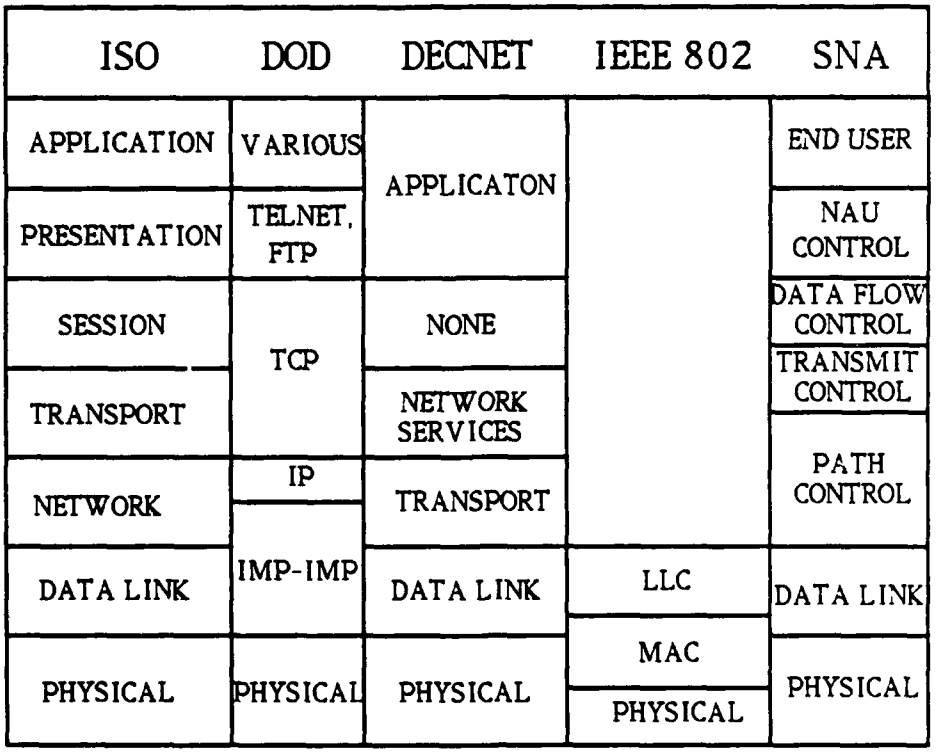


Figure 1-3. COMMUNICATION PROTOCOL SUITES

their productivity by defining the communication tasks to be performed in a consistent fashion. In keeping with sound software engineering principles, the architecture should decompose the problem space into manageable units (modularity), keep task functionality in the same unit (strong cohesion), and minimize the impact on surrounding units when changes are made (loose coupling). [23] Beginning in 1977, ISO established an architecture exhibiting these characteristics, collectively known as "layering", -- the Open Systems Interconnection (OSI) model (see figure 1-4). When considering all of the layers within the OSI model as a single entity the term "protocol suite" will be used in this thesis.

7	APPLICATION LAYER
6	PRESENTATION LAYER
5	SESSION LAYER
4	TRANSPORT LAYER
3	NETWORK LAYER
2	DATA LINK LAYER
1	PHYSICAL LAYER

Figure 1-4. OSI REFERENCE MODEL

Since its formulation, OSI has been used extensively in the design and implementation of computer networks and only those networks adhering to this model (not necessarily in exact detail, but in theory) will be considered in this thesis. An assumption of familiarity with the OSI model and the layering concept is made at this point. A detailed description may be found in ISO's International

Standard 7498. [24] The layering concept not only provides the foundation for independent network implementation, but also for internet construction.

1.6 Internets - How

Whereas independent networks implementations are concerned with each layer within a protocol suite, the internet designer assumes there are complete suites already in place. The objective of any type of internetworking approach must be the creation, at some point in the protocol suite, of a layer of commonality. The layering decision to be made in this context is whether to use the existing protocol suites as they are and implement a rather intelligent gateway between them, or add another layer, identical in each suite, to perform this task.

The next two sections provide an overview of these two techniques, a more detailed discussion will be given in chapter five.

1.6.1 Using Existing Protocol Suites

If the decision to use existing protocol suites is made, the internet designer must then determine the degree of compatibility between subscriber subnet services. If the internal services of each subnet are identical, internet gateways serve as simple relay stations to forward traffic between subnets. [25]. An example of this type of internet is found in CCITT's X.75 standard. [26] When the degree of service compatibility is anything but identical some type of translation mechanism must be provided within gateways before traffic may be passed on to subsequent subnets. As the degree of compatibility decreases, translation complexity increases. Both methods (relay and translation) may be viewed as "stepwise", or "hop-by-hop" approaches to internetworking in that the services of each subnet are used in their present form with forwarding and/or

translation of traffic being performed at the gateway. [27], [28] An obvious advantage of this approach lies in the fact that existing subnet protocols require no changes. However, as subnets become so incompatible as to make the translation mechanism extremely difficult, or perhaps impossible, an alternative approach is necessary.

..6.2 Using an Additional Layer

The translation mechanisms just discussed are usually implemented in a proprietary fashion, thus reducing the chances of one translation attempt successfully connecting subnets of another. For this reason, protocol translators have not gained much support in the internetworking community. A more popular approach consists of adding a common protocol, known as the "internet protocol", to each of the subnet hosts and internet gateways. The internet protocol is not directly involved with the internal operation of each subnet, but is necessary to bridge the differences in subnets, thus allowing traffic to be exchanged throughout the internet. [29] The internet protocol method is sometimes called an "endpoint" approach to internetworking since each communicating end practices the same protocol within its protocol suite.

Having considered the options, we may now choose the internetworking approach to be used in thesis.

CHAPTER 2

Internetwork Layer Determination

By placing internetworking functionality within a single layer the same benefits as those discussed with regard to the OSI model (modularity, strong cohesion, weak coupling) are realized. The next internetworking decision to be made is that of selecting a particular layer to use. Operational characteristics and services offered by each layer must be considered when making this decision. The next section will demonstrate the (dis)advantages of each layer, from an internetworking perspective, and will select one to form the architectural basis of this thesis.

2.1 Layer Analysis

It is important to distinguish between OSI layers belonging to the CSS and those known as "end-to-end" when choosing an internet layer. Layers one, two, and three, or "lower layers", are part of the CSS and as such are concerned with purely communication aspects of the (inter)network. The remaining end-to-end layers consist of the transport layer and layers five, six, and seven, or "upper layers". The three upper layers are concerned with using data communicated through the CSS for application-specific purposes, while the transport layer has both communication and application-specific properties. [1] This dual nature of the transport layer, represented in figure 2-1, will prove to be very important in determining an internet layer. The operational difference between CSS and end-to-end layers is that each intermediate stop along the communication path (in the case of packet-switched networks) must have active CSS layers in order to keep messages moving toward their final destination, while the end-to-end layers are active only at source and destination hosts. [2] An earlier assumption was made

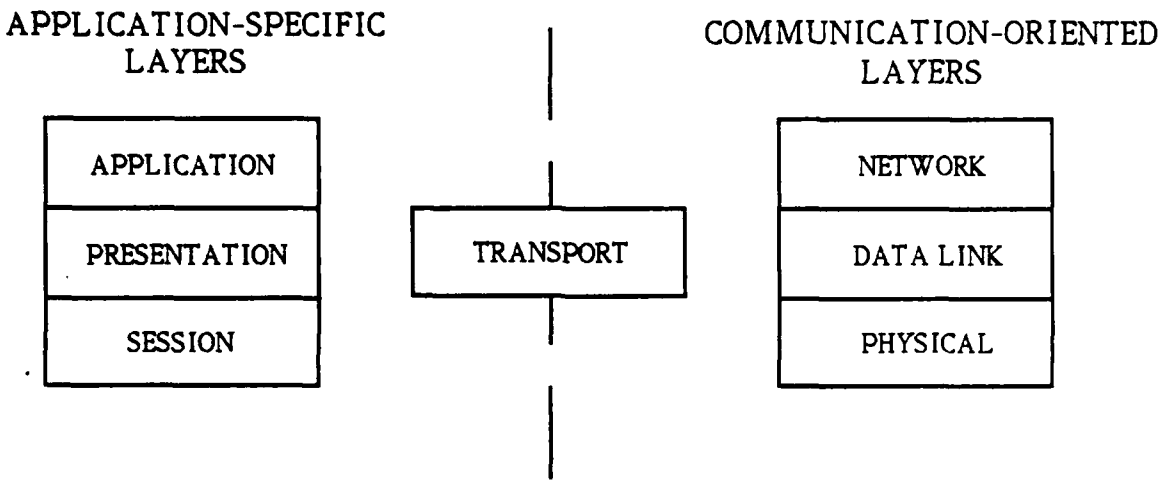


Figure 2-1. DUAL NATURE OF TRANSPORT PROTOCOL

in chapter one regarding the type of service provided by the CSS, namely a connectionless service. Since the layer used for creating an internet must permit information of end-to-end significance to pass through it, layers one, two, and three are ruled out.

Although the most popular method in use today, the addition of a common internet protocol layer will not be used due to the insistence that every host within every subnet implement it. An architecture based upon little or no internal subnet modification is preferred. The preceding discussion leaves layers four through seven as possible internetwork layers.

Concerning layers five through seven, the ability to specify particular functions, generic to any particular implementation, becomes rather vague. These layers, "..are so diverse that an all-embracing protocol conversion which retains the defined end-to-end conditions for every layer is not feasible." [3] It is unlikely that this situation will change since these layers provide the specialized services computer users demand and as such were not designed with

interoperability in mind. Thus we have narrowed the field of possible internetwork layers to one, layer four of the OSI model -- the transport layer. Justification for selecting this layer will be given next.

2.2 Internetwork Layer Justification

Somewhat of a dichotomy exists in the design of internets in that the benefits of interconnecting multiple networks are desired, while at the same time preserving the autonomy of each network as much as possible. [4], [5], [6] For the following reasons, use of the transport layer supports both sides of this argument.

2.2.1 Independence From Network-Unique Features

From both the intranet and internet perspective, upper layer users of the transport protocol are not concerned with the type of CSS used below them. The transport layer serves to shield any peculiarities of CSS operation from upper layer protocols. [7] In fact, network administrators could swap one set of CSS layers for an entirely different one and the upper layers would have no knowledge of the change.

2.2.2 Transmission Optimization

Although overlooked many times, this function of the transport layer may result in an otherwise simple transfer of information being delivered poorly. There are two types of optimization performed at this layer. First, there may be more than one transport protocol to choose from (each one based upon a different kind of CSS). [8] The transport user specifies performance characteristics by setting optional parameters at "connection-request" time made available by the transport service provider. [9] These parameters may include throughput, transit delay, error rate, failure probability, and transmission priority

level [10], [11] and are used to determine the appropriate protocol to invoke. The second optimization comes through passing these transport user/provider parameters on to the CSS where they are compared against its "quality-of-service" (QOS) parameters. The transport layer, therefore, attempts to bridge the gap between what the transport user wants and what the CSS can provide. [12]

2.2.3 Information Enveloping

Although not unique to the transport layer, any information sent to the transport layer is treated as raw data and as such is simply "enveloped" within the transport header and passed on to the next layer. There is no restriction on content, format, or coding of the information, nor is there ever a need to interpret its structure or meaning. [13]

2.2.4 Own Addressing Scheme

As mentioned above on optimization, the transport layer may actually consist of multiple protocols. These protocols, in turn, are supported by multiple "entities" that implement the services of their protocol and communicate with "peer" entities in other computers on the internet. From an earlier assumption, all transport communication involves an explicit connection. Therefore, a mechanism for addressing a specific entity from all possible transport entities within the layer must be provided. This is accomplished through the use of "connection endpoint" (CEP) identifiers and "service access points" (SAPs) as illustrated in figure 2-2. [14] At every layer there exists a pool of CEP identifiers, the structure of which is known by other CEP identifiers throughout the internet at the same layer. As entities request connections with peer entities located in the internet, a CEP identifier is assigned to the requesting entity and it is through this identifier that communication actually takes place. One more

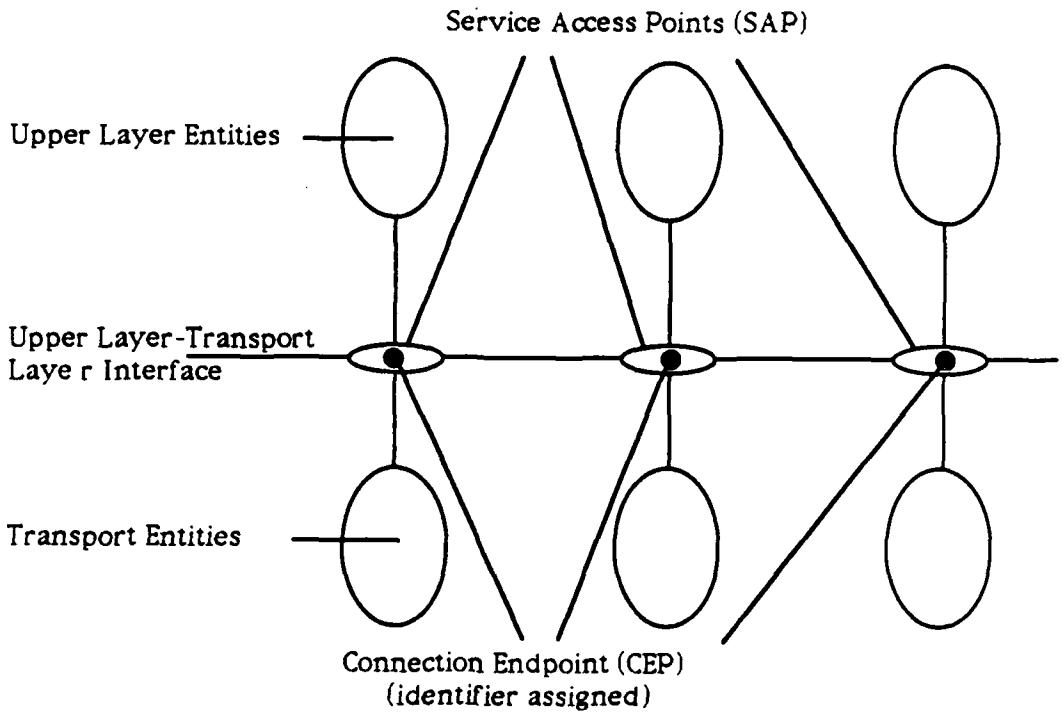


Figure 2-2. ENTITIES, CEPs, and SAPs

level of indirection comes from the fact that only at layer one in the ISO model does a physical connection exist. Therefore, **logical** connections (via CEP identifiers) must be passed down the protocol suite across layer boundaries. The boundaries are penetrated, thereby making lower layer services available, at SAPs. As CEPs are allocated they are associated with a particular SAP. [15], [16]

To summarize, entities are tied to CEPs which are, in turn, tied to SAPs. As references to a particular CEP enter a layer, a mapping function directs them to the appropriate SAP. Although not unique to the transport layer, the SAP/CEP/entity association process does provide the ability for the endpoints of a transport protocol-based internetwork conversation to uniquely identify each other among all others in an internetworking environment.

2.2.5 End-to-End Reliability/Correctness

The previous justifications have been important for implementation reasons, but from an architectural perspective the single most important reason for choosing the transport layer is found in the fact that it, and only it, is responsible for the **reliable, error-free** exchange of information between separate computer systems within a(n) (inter)network. [17], [18] To better understand this let us distinguish between communication **signals, data, and information.**

At layer one of the OSI model we find nothing more than raw electronic **signals** being propagated along a physical transmission media. It would be impossible to connect heterogeneous subnets at this level since these signals carry no intrinsic meaning. Moving through layers two, three, and four the signals are given meaning by attaching special header and/or trailer sequences, resulting in the logical grouping of signals into **data.** Another transformation, illustrated in figure 2-3, takes places as data leaves layer four destined for the upper layers where it becomes reliable, application-specific **information.** Since it is at the application-specific layers that meaningful work is accomplished, it makes sense to place the internetworking functionality as close to these layers as possible -- at the transport layer.

2.3 Summary

With regard to the connectivity/autonomy tradeoff introduced earlier, the use of layer four as an internetwork layer allows each subnet to exercise its own CSS protocols completely independent from other subnets, while also providing error-free, application-specific information to upper layer protocols. The services found at layer four allow many **different** applications to use the same transport protocol, much like mopeds, passenger cars, motor homes, and 18-wheelers all use the highway system for a common conveyance. [19]

The sections of this chapter have presented solid evidence for using the transport layer in an internetworking role, but this is not to imply an absence of problems in doing so. The next chapter will cover specific services provided by the TCP and TP 4 protocols and will point out areas of inconsistency between their respective implementations.

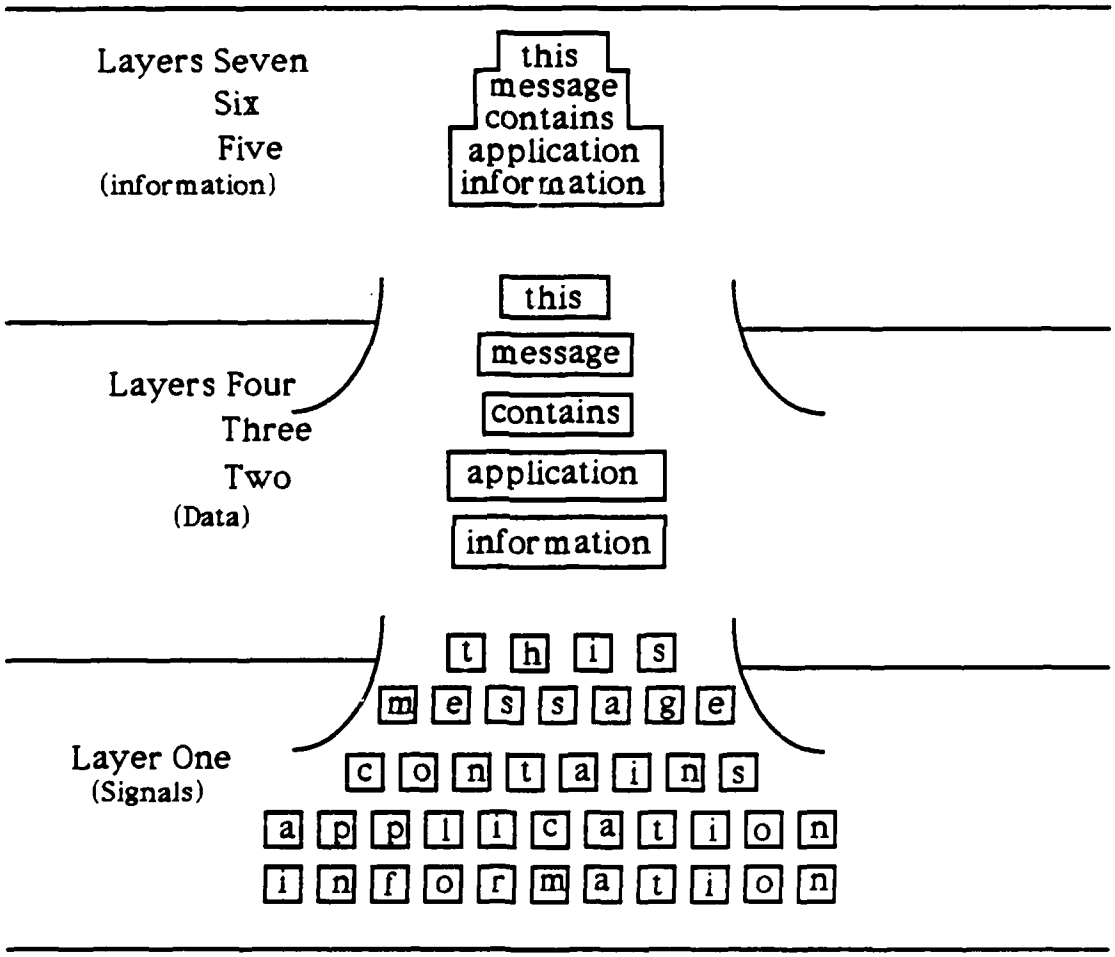


Figure 2-3. SIGNAL-DATA-INFORMATION TRANSFORMATION

CHAPTER 3

Transport Protocol-based Internet Services

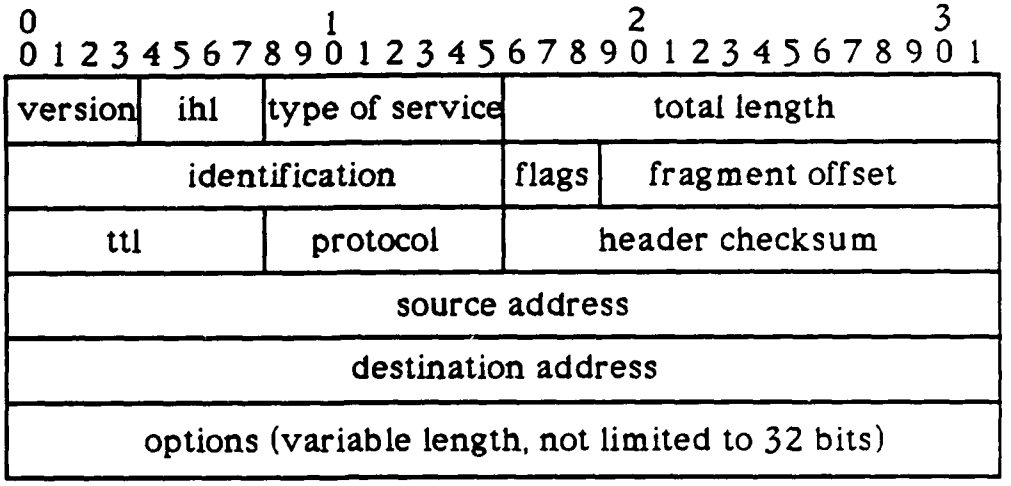
The services demanded of a heterogeneous, transport layer-based internet fall into several groups. The first includes services traditionally associated with transport protocols, while another contains services normally performed at the network layer. Since the information used in performing services of the latter is found only in the message's network envelope, the architecture of this thesis will assume access to this information as well as that found in transport layer envelopes. Figures 3-1a and 3-1b (on pages 21 and 22, respectively) illustrate the transport and network envelopes for TCP and TP 4-based messages; the fields within these envelopes will be referred to throughout this chapter and during the presentation of the architecture in chapter four. The third and final group consists of services required only when dealing with heterogeneous subnets. The objective of this chapter will be to point out the similarities and differences between TCP and TP 4-based networks in providing services from each of these groups.

3.1 Traditional Transport Services

The single most important factor determining the services of a transport protocol is the reliability of the underlying layers.[1] In order to facilitate standardization efforts, ISO has defined the following levels of CSS performance : [2]

> Type A : a CSS with an acceptable residual error rate and an acceptable rate of signaled failures (completely reliable),
>
> Type B : a CSS with an acceptable residual error rate, but unacceptable rate of signaled errors (less reliable),

TCP-based network envelope



TCP-based transport envelope

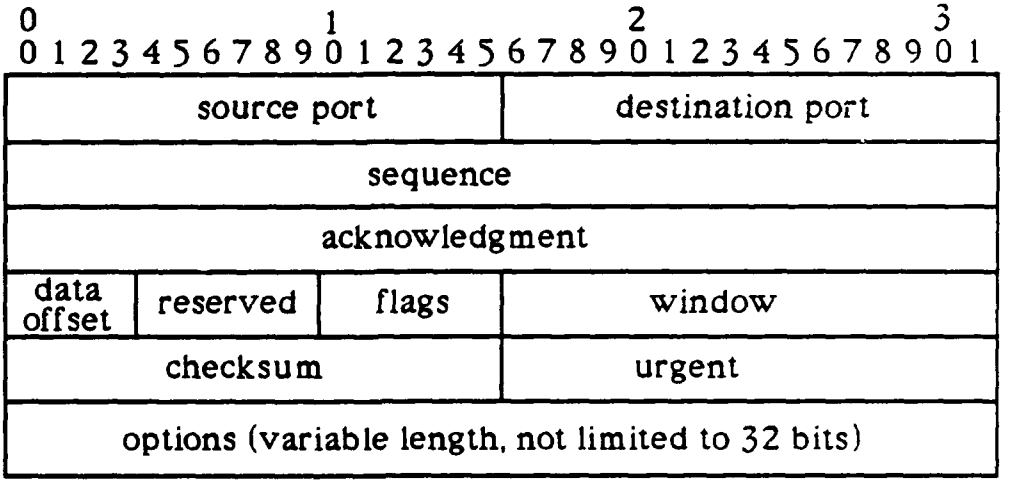


Figure 3-1a. TCP-BASED ENVELOPES

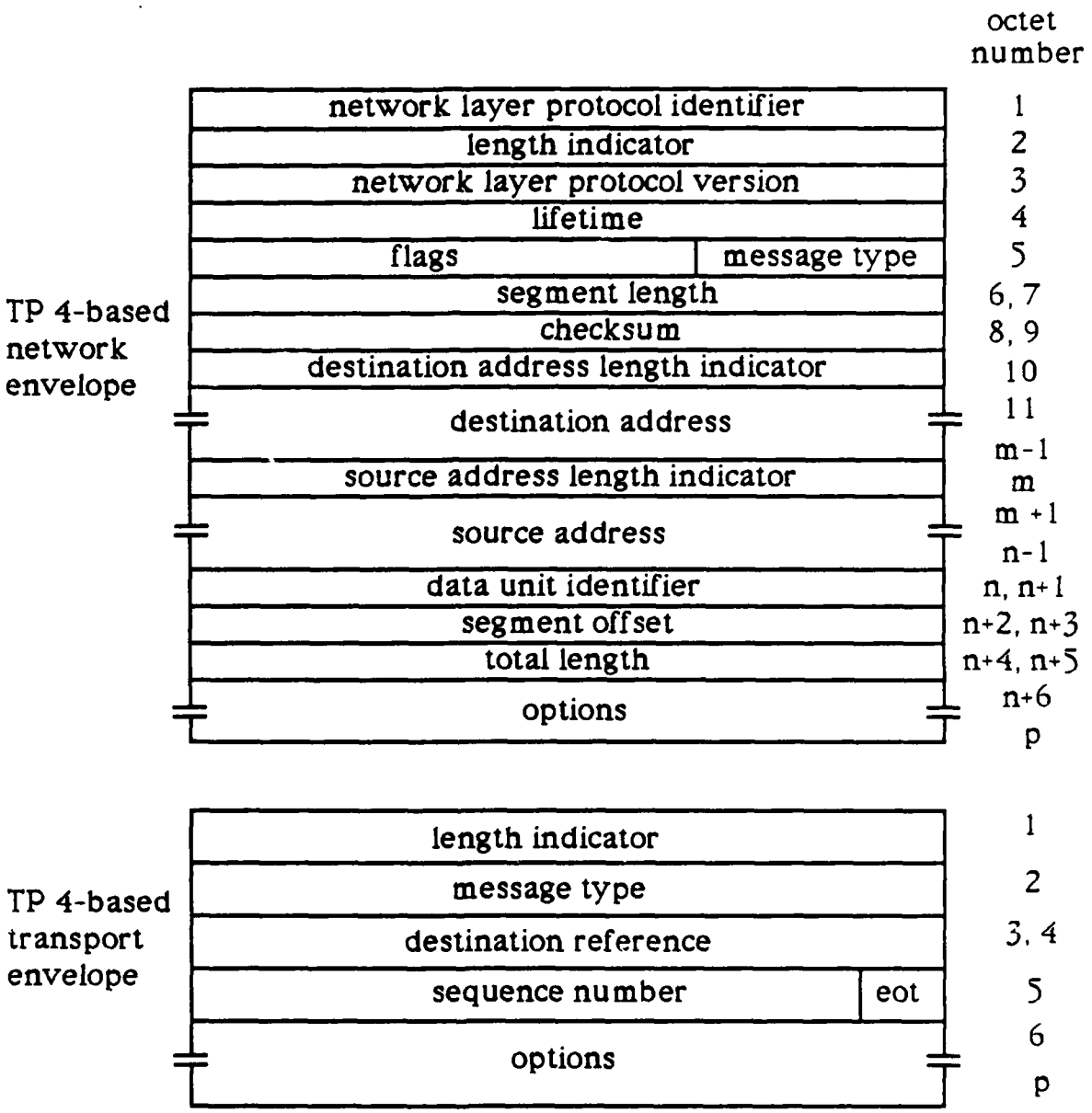


Figure 3-1b. TP 4-BASED ENVELOPES

> Type C : a CSS with residual error rate not acceptable to the transport user (completely unreliable), where **signaled errors** are detected and not corrected by the network layer, but **are reported** to the transport layer. **Residual errors** are detected, not corrected by the network layer and **are not reported** to the transport layer.

Our connectionless CSS falls in the type C category and, therefore, requires considerable sophistication on the part of the transport layer. To compensate for the residual errors of a Type C CSS, yet still utilize its capabilities as efficiently as possible the following services are required at the transport layer : [3], [4]

- Connection Management
- Transparent Message Delivery

Services, as discussed thus far, are merely high level abstractions of desirable transport layer characteristics. The vehicle used to supply these abstractions throughout layers of the OSI model are the communication entities introduced in chapter two. These entities, in turn, consist of specific implementation "mechanisms". Mechanisms may be thought of as the algorithms or data structures developed by communication software engineers. It is the **mechanisms** that give life to transport services and will be explained in the following sections.

3.1.1 Connection Management Mechanisms

Before messages may be passed between transport users a connection must exist. The transport layer is the first connection oriented layer in protocol suites assumed for this thesis and is, therefore, responsible for managing these connections.

In establishing a connection, essential initialization information must be exchanged between each end. The mechanism used to ensure successful performance of this task in both TCP and TP 4 is illustrated in figure 3-2 and is known as the "three-way handshake". [5], [6] The figure shows each end, in

turn, requesting permission to open a connection with the other, along with an indication as to the willingness of each to accept the requests. Under normal circumstances the process proceeds as shown, but due to the unreliable nature of the CSS, delayed and/or duplicate messages from previous connections may be mistaken for original ones. Rather than explain the possible erroneous scenarios, suffice it to say that the three-way handshake results in unambiguous connections between transport user entities.

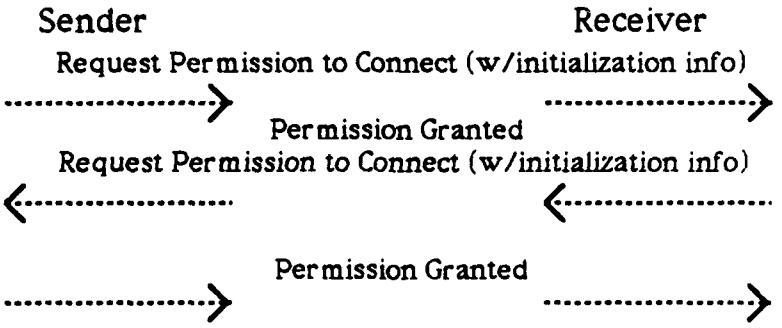


Figure 3-2. CONNECTION ESTABLISHMENT

Connection termination is also performed using the three-way handshake mechanism, as illustrated in figure 3-3. Once connected, transport users exchange information using a collection of reliability mechanisms to be explained next.

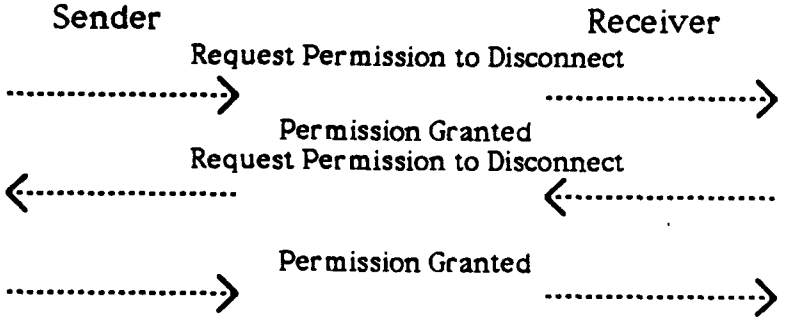


Figure 3-3. CONNECTION RELEASE

3.1.2 Transparent Message Delivery Mechanisms

In our discussion on transport layer justification (see chapter two) the reliable, error-free nature of transport protocols was given as the most important reason for using it as the internetwork layer. Consistent with its importance, the transparent delivery of information between two transport users also requires the most from a transport protocol. The following mechanisms are needed to provide this service : [7], [8]

- Positive Acknowledgement with Retransmission
- Flow Control Windows
- Duplicate/Out-of-order Detection
- Checksum

Before describing each of these mechanisms the notion of "sequencing identifiers", a design decision of primary importance to each of the mechanisms except the checksum, will be explained.

Regardless of how a protocol structures its messages some type of identification scheme must be agreed on among communicating members. TCP and TP 4 use integers as sequencing identifiers, or sequence numbers. As each message is sent, it is given a unique number (unique within the expected lifetime of any message belonging to the same connection) by the sender. The receiving end may then use this unique number in determining whether the message has/has not arrived in correct sequential order (positive acknowledgement/out-of-order detection), has already been accepted (duplicate detection), or is not currently acceptable (flow control windows) to the receiver.

3.1.2.1 Positive Acknowledgement with Retransmission (PAR)

As information is exchanged between peer entities there must be, at some point in the protocol suite, the means for providing feedback to the sender as to the receiver's success (or lack thereof) in actually obtaining the information. In

both TCP and TP 4 the PAR mechanism has gained popularity in performing this task. [9], [10] The basic notion of this mechanism consists of a receiving transport entity sending a positive acknowledgement (ACK) to the sender, in the form of a sequence number, for each message it accepts. This has the effect of telling the sender that the next message it sends should contain the sequence number contained in the ACK. As might be expected, initial synchronization of sequence numbers is imperative for proper PAR operation. The three-way handshake mechanism previously discussed ensures synchronization is obtained and each transport user entity maintains sufficient state information to provide correct sequence number interpretation throughout the duration of their connection. When irregularities occur in message/ACK sequencing or when messages/ACKs are not delivered, the retransmission aspect of the PAR mechanism is invoked.

Messages and their associated ACKs are subject to the unreliable CSS. For this purpose the PAR mechanism provides a "timeout" for each outgoing message. As each message is sent, the timeout period begins to count down. If the timer expires before the message is acknowledged, or if the ACK received is for an out-of-sequence message, the sender once again transmits it. One reason for lost messages/ACKs comes from the possible exhaustion of resources at the receiving end. The next mechanism is intended to prevent such a situation from occurring.

3.1.2.2 Flow Control Windows

The flow control mechanism provides the ability for the receiving end of a connection to "throttle" the amount of traffic coming from the sending end. Various mechanisms are in use, but the most popular (used by TCP and TP 4) is based upon a "credit allocation" algorithm. [11] At any instant each transport entity has two windows -- one specifying how many messages it is willing to

receive (receive window) and another indicating the number of unacknowledged messages it may have in transit to the other end (send window). Certain fields within protocol envelopes contain the values used to update these windows.

Flow control windows are closely related to the PAR mechanism, serving as a restriction on the range of acceptable sequence numbers. Another mechanism associated with both flow control and PAR is that of duplicate/out-of-order detection.

3.1.2.3 Duplicate and Out-of-order Detection

The ability to detect duplicate and out-of-order messages is actually nested within other transparency mechanisms and the sequence number-based state information maintained by each transport entity. When messages with sequence numbers previously ACKed arrive they are treated as duplicates and not passed on to higher layer protocols. Those messages that fall within an entity's receive window, but not in correct sequential order may be discarded or buffered until intervening messages arrive, depending on the protocol implementation. [12] Thus far only sequencing problems have been dealt with. The transport protocol must also detect damaged messages.

3.1.2.4 Checksum

Although not the only technique used for transport protocol error checking, TCP and TP 4 use a software checksum due to its relative simplicity, yet sufficient error-detection properties. [13] The sending transport entity computes an initial checksum value and sends it along as part of the message. If the receiving entity computes a different checksum value, the message is discarded and the PAR mechanism at the sending end will retransmit it.

Connection management and transparent delivery mechanisms provide the functionality traditionally associated with transport layer protocols. The next portion of this chapter deals with services typically found in network layer protocols.

3.2 Traditional Network Services

Although performed at the network layer on a subnet-by-subnet basis, the ability to distinguish an object among all addressable objects (name-to-address resolution) and the partitioning of large messages into smaller, CSS-manageable messages (fragmentation/reassembly), is also inherited by the transport layer when it serves as the internet layer. As might be expected, the lack of transport layer familiarity in performing these services introduces a significant amount of incompatibility to internet operation.

3.2.1 Name-Address Resolution

By far, this issue introduces the greatest degree of confusion when connecting heterogeneous networks. There are as many ways of naming resources on a network as there are networking vendors. Only when internetworking became an important issue did the lack of commonality between these schemes become apparent. Many of the problems arising in this area are rooted in the question of subnet autonomy vs internet functionality -- each network's name/address space should be preserved to the greatest extent possible, yet global agreements bring about improved internet operation. [14] Since the basic operation of any (inter)network depends on its ability to uniquely reference an object with whom communication is desired, techniques to provide this function must be available or all else is of no value. The final problem area to be

discussed deals with the ability of an internet to accommodate different sized messages.

3.2.2 Message Fragmentation/Reassembly

Simply stated, networks, including those based upon TCP and TP 4 protocols, impose different sizes on their messages. There are various reasons behind such limitations, including : [15]

- available bandwidth
- restrictions on buffer size within network hosts
- desire to reduce error characteristics
- desire to establish some sort of "fairness doctrine" among hosts
- compliance with protocol standards

Whatever the cause, an internet architecture must include mechanisms for breaking up and reconstructing messages in accordance with subnet size constraints, while preserving the content of the original message.

The last group of services to be discussed have no corresponding implementation in stand-alone TCP and TP 4 networks. In fact, it is only because of the heterogeneous nature of this thesis' internet that they are mentioned at all.

3.3 Heterogeneous Internet Services

Similar to the services of section 3.2, those of this section are routinely performed by individual subnets. However, when attempting to cross heterogeneous subnet boundaries, mutually exclusive or largely incompatible allocation mechanisms for these services suggest potential problems.

3.3.1 Sequence Preservation

At the beginning section 3.1.2 the idea of message sequencing was introduced. The only similarity between the TCP and TP 4 sequencing scheme is that they each have one. TCP has chosen to number each outgoing octet (eight

bits), with the sequence number of the first octet in the message being treated as the sequence number for the entire message. [16] Sequence numbers for initial messages over newly created TCP connections are assigned by an "initial sequence number generator". The sequence number for the next message is determined by adding the sequence number and message length (in octets) of the initial message. The same process is then used for subsequent messages sent over the connection. TP 4 sequence numbers, on the other hand, are independent of the length of previous messages. Each message is simply assigned a number in ascending order. [17] When exchanging messages across a TCP/TP 4 boundary, obvious sequencing discrepancies will exist, as illustrated in figure 3-4.

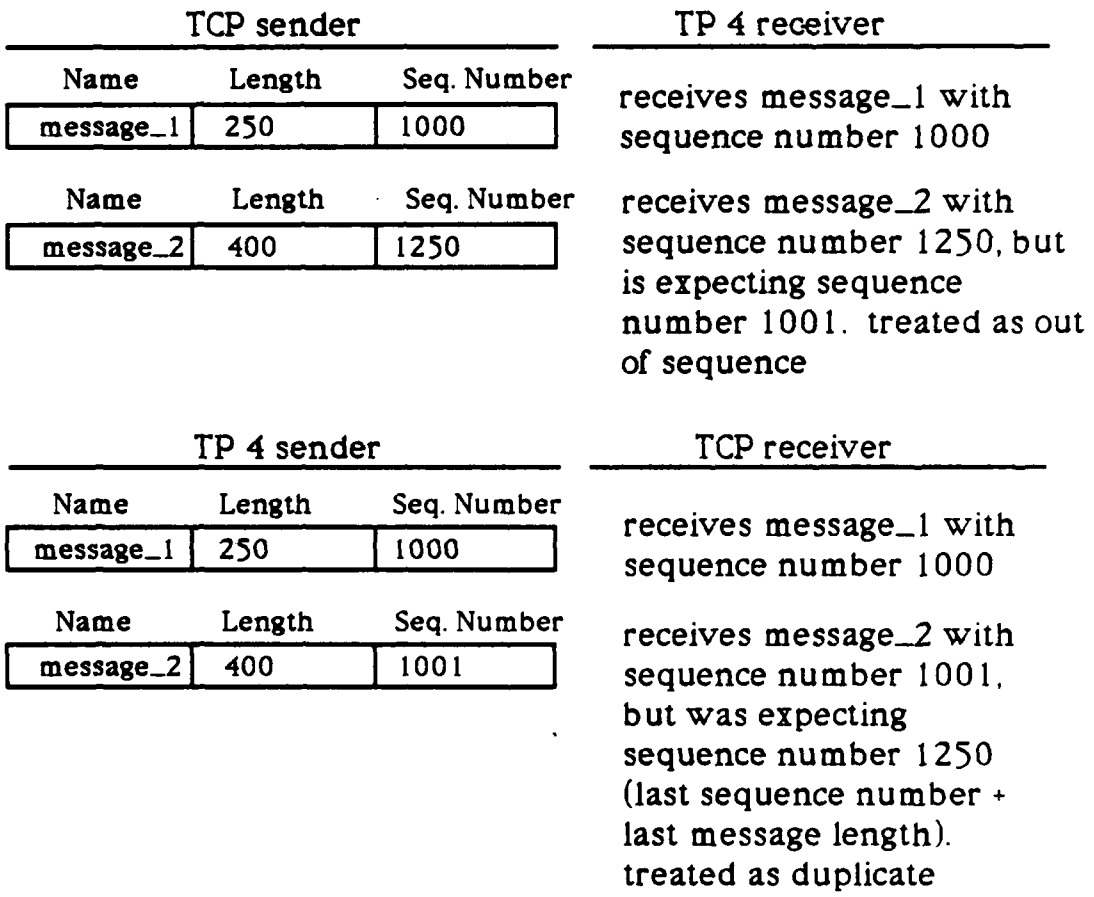


Figure 3-4. SEQUENCE NUMBER EXCHANGE

Two consecutive TCP messages may have sequence numbers of 1000 and 1250 (the first contains 250 octets). If these messages are merely handed off to a TP 4-based host, it will see a void of 250 messages. Conversely, a TP 4 message having sequence number 1000 and a length of 250 octets followed by another message with sequence number 1001 will result in the second message being ignored as a duplicate by a TCP host.

Just as with sequence numbers, the quantity and quality of optional parameters offered by one network may differ greatly from another. When crossing heterogeneous subnet boundaries, efforts must be made to preserve as many as possible.

3.3.2 Option Preservation

Network and transport envelopes in both TCP and TP 4-based networks contain a variable length part for optional parameter selection. These parameters are used to carry special information or to specify various constraints that must be met when transmitting messages between connection endpoints. The (non)selection of these parameters is entirely up to the transport user. Messages may have zero or more parameters, with each being completely independent from any other. In TCP-based messages, only one option of any use -- maximum message size, is contained in the transport envelope. [18] The network envelope contains the remainder of the selectables available to TCP users as shown in table 3-1. [19] As for TP 4-based messages, both envelopes (see table 3-2) are capable of carrying many options. [20], [21]

The problem with option selection, from an internetworking point of view, comes when a message from one subnet is destined for, or must pass through, another subnet providing a different set of options. The architecture used must

TABLE 3-1. TCP-BASED OPTIONS	
TRANSPORT ENVELOPE OPTION	NETWORK ENVELOPE OPTIONS
Maximum Message Size	Padding Priority Delay (high/low) Throughput (normal/high) Reliability (normal/high) Security Source Routing Route Recording Stream Identifier Internet Timestamp

TABLE 3-2. TP 4-BASED OPTIONS	
TRANSPORT ENVELOPE OPTIONS	NETWORK ENVELOPE OPTIONS
Transport SAP Identifier Checksum Maximum Message Size Version Number Security Alternate Transport Protocols Acknowledge Time Priority Throughput (desired/minimum acceptable) Error Rate (desired/minimum acceptable) Transit Delay (desired/maximum acceptable)	Padding Security Source Routing Route Recording Sequencing vs Transit Delay Congestion Experienced Transit Delay vs Cost Error Rate vs Transit Delay Error Rate vs Cost

provide some type of matching function to determine those options that make it past the subnet intersection.

All that has been said up to this point has but pointed out the need for some type of internetworking solution. Several possibilities have been explored and some have reached the operational phase. The next chapter will present the architecture for this thesis and show how it resolves the incompatibilities covered in this chapter.

CHAPTER 4

Internet Architecture

The architecture proposed in this thesis is best referred to as a "meta-protocol" for connecting computer networks. Just as meta-programming languages, such as Backus-Naur Form (BNF), allow different high-order languages to be described in the same format, a meta-protocol permits more than one communications protocol to be encoded in a common manner. The strength of this approach to internetworking lies in its answer to the question posed several times in this thesis -- subnet autonomy or internet functionality. The meta-protocol provides favorable responses to both sides of the argument. Internet message exchange is supported through gateway devices where conversion to/from subnet-specific transport protocols and the meta-protocol is performed. As with any internetworking attempt there must be some level of agreement between those desiring connectivity. With regard to the meta-protocol consensus must be reached in the following areas :

First, each subnet must understand the format of the meta-protocol, thereby allowing the protocol conversion software, located at the internet gateways, to be written. The gateways for the meta-protocol architecture will actually consist of two halves, one belonging to each subnet they connect. [1] Using this concept, message exchange through a gateway proceeds in the following manner. Messages arrive at the gateway half responsible for converting it to the meta-protocol format (exit half). The conversion is performed and control is transferred to another gateway half (entrance half) where the meta-protocol format is converted into a subnet-specific format. This process continues until the host corresponding to the message's destination address is found to reside on subnet directly connected to an entrance gateway half.

Second, a common naming convention for all internet hosts must be adopted and finally, some type of "directory" service must be available to internet hosts and gateways that performs a mapping from subnet-specific names to internet names and visa versa. Each of these areas will be treated in answer to the problem areas discussed in chapter three. A topological abstraction of a meta-protocol-based internet will next be presented.

4.1 Logical View of the Architecture

Figure 4-1 illustrates the logical view of computer networks connected via meta-protocol gateways. As was mentioned previously, this architecture is based upon the catenet model (see figure 1-2); the similarities are obvious. To understand how this logical structure provides transparent connectivity among heterogeneous internet hosts, further explanation of the message exchange process and the part played by meta-protocol gateways is required.

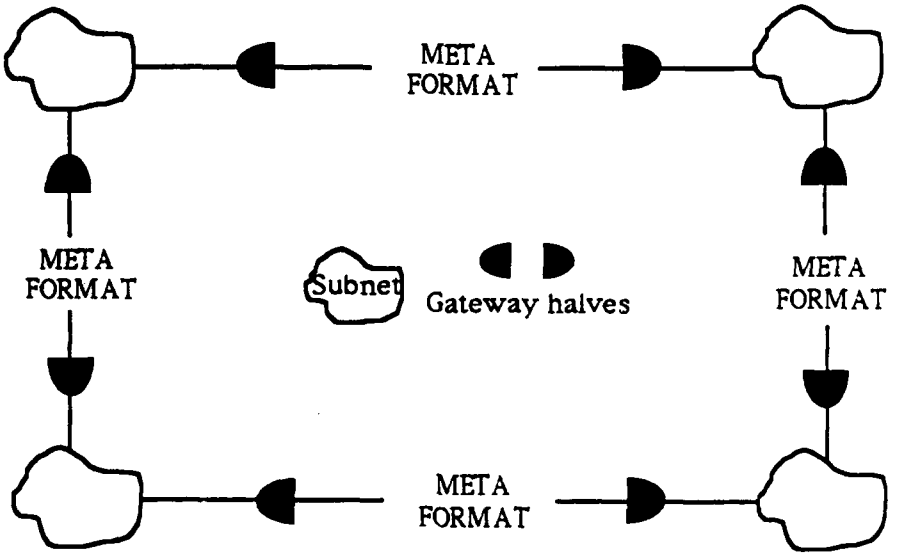


Figure 4-1. META-PROTOCOL INTERNET

"The fundamental role of the gateway is to terminate the internal protocols of each network to which it is attached while, at the same time, providing a

common ground across which data from one network can pass into another." [2] Except for the fact that they cannot be explicitly addressed, meta-protocol gateway halves are treated as any other internet host. Internally, two sets of functions, as illustrated in figure 4-2, are implemented in each half. The first consists of the network-unique protocols up to and including the transport protocol. The other, supplied by each subnet connected to the gateway, is responsible for the conversion to/from the meta-protocol format. The conversion from the network-unique format to the meta-protocol format is accomplished by extracting certain pieces of information (to be identified as the chapter progresses) from the network and transport layer envelopes and putting them in appropriate meta-protocol fields.

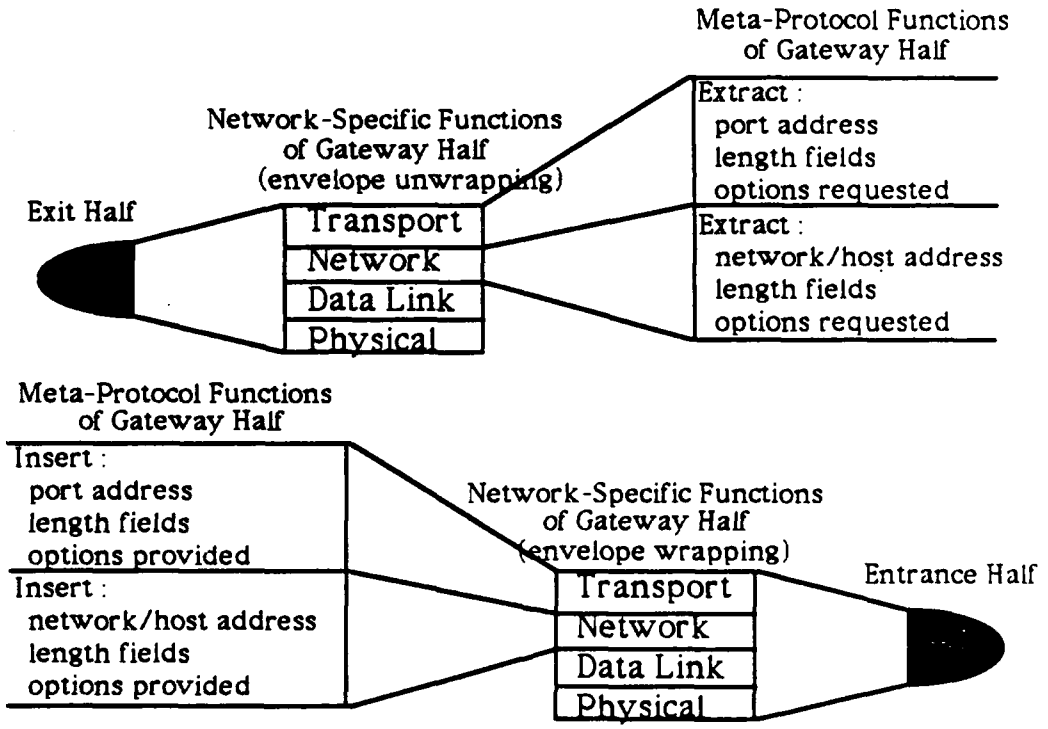


Figure 4-2. FUNTIONS OF META-PROTOCOL GATEWAY HALVES

The conversion into a network-unique format is performed by taking entries from the meta-protocol-based data structure and inserting them in the correct locations of network/transport layer envelope. It is important to note that only **critical** fields within protocol envelopes are passed from exit to entrance halves; all others are supplied through the normal operation of each subnet's protocol suite.

The distinguishing feature of this particular architecture comes through it placing the conversion software at the transport layer in the protocol suite. By coupling subnets at this layer is it possible to "splice" together **actual** connections (hops) from different protocol suites, the result being a single **virtual** connection, exhibiting the same properties of reliable, error-free transport delivery as explained in chapter three. CCITT has chosen a similar technique in its X.75 internet technique, but the difference between it and the meta-protocol approach lies in the ability of X.75 to connect **only** CCITT X.25-based subnets. [3] As in the OSI reference model, the organization of the meta-protocol is based upon the layering concept. Figure 4-3 illustrates the logical division of the meta-protocol into five separate sub-layers. Each sub-layer is responsible for supplying a portion of the overall internet functionality. Referring to the problem areas of chapter three, a one-to-one correlation exists in this layered architecture. The figure has also been arranged in somewhat of a sequential fashion. As each sub-layer's role is defined in the remaining sections of this chapter, the chosen sequence will become evident.

Since this thesis forms an architecture, the implementation techniques used to support it are of no concern. Any data structures or algorithms suggested in this chapter are for illustrative purposes or will be used in the simulation described in chapter six.

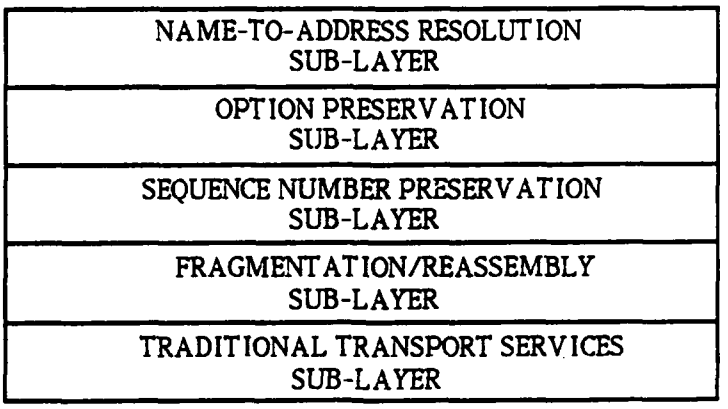


Figure 4-3. META-PROTOCOL SUITE

4.2 Name-to-Address Resolution Sub-layer

It is at this point in the meta-protocol that there exists the greatest potential for degradation of subnet autonomy. The technique used for this purpose must be, at the same time, supportive of internal addressing schemes and powerful enough to specify any object in the internet. It would be ridiculous to assume, however, that such functionality is obtained without some prior agreement among participating subnets. The assignment of internet addresses cannot occur in an ad hoc fashion; there must be an authority responsible for allocating addresses and ensuring address uniqueness throughout the internet. With regard to address assignment, the format used will determine, to a large degree, the success (or lack thereof) of internet operation. There are two alternatives -- flat and hierarchical. When making a decision as to the preferred technique, two factors must be considered. First, the ability to ensure uniqueness throughout all hosts on the internet and second, any inherent information contained in the address that may assist in the routing function. [4]

4.2.1 Flat Address Space

The social security numbering system provides an example of addresses based on a flat format. Regardless of where the addressable object is located, the same pool is drawn from when assigning values. With respect to the first of our criteria for format determination we find a single authority responsible for every address assigned. Before any object may be added to any subnet within the internet, this authority must be consulted. [5] As for addressing information carried by the address itself, there is none. Just as two consecutive social security number assignees may live on opposite coasts of the United States, two internet objects with consecutive addresses may have no relation to one another with respect to geographic location. [6] Due to these weaknesses, the meta-protocol will adopt a hierarchical addressing format.

4.2.2 Hierarchical Address Space

In direct contradiction to a flat address space, a hierarchical form allows for multiple allocation authorities, while still ensuring address uniqueness among all subnets. It also allows the address to convey a significant amount of routing information without having to interpret the entire address. A caveat must be added when speaking of independent address allocation authorities -- only within their assigned subnet or group of subnets are they allowed to assign addresses. As some point a supreme addressing authority (internet authority) is still required, but using a hierarchical format reduces the responsibility of this person/organization to that of breaking the entire internet into smaller "domains". These domains are, in turn, administered by domain-specific authorities. [7] As for information carried by the address itself, the agent responsible for performing the routing function (gateways for the meta-protocol) need only look at the outermost level of the address in order to obtain a "very strong hint" as to where

the message need be sent. [8] In both TCP and TP 4 this outer level corresponds to a network number. When a TCP or TP 4-based internet router receives a message it first looks at the network number, if this number is the same as the network portion of the router's address the message is meant for a host on a network directly connected to the router. Otherwise, the router consults its routing table to determine which other routers may receive the message and forward it toward its ultimate destination. [9] When arriving at the destination network the next level of addressing is interpreted to determine the local host that should receive the message. The following hierarchical addressing scheme, suggested by Xerox Corporation in [10], has been modified to fulfill the requirements of the meta-protocol architecture.

4.2.3 Meta-Protocol Name-to-Address Resolution

As these terms will be used throughout this section it will be necessary to define exactly what is meant by a "name" and an "address". A name is a symbol, usually presented in the form of a human-readable character string. Such a construct is merely for the benefit of humans and has no meaning to the internal operation of the internet except as a key word used to perform address mappings against. [11] The particular scheme being proposed for this architecture will use a three-tiered naming convention consisting of the following components :

> User Name : This portion of the address does have a corresponding address mapping, rather it is used by protocols above the transport layers (e.g. the name of the mailbox where incoming mail should be deposited). The only restriction placed on these names is that they be unique within hosts. There may be duplicate user names on different hosts.
>
> Host Name : Using a hierarchical format, this component is assigned by independent domain authorities. The only restriction placed on these names is that they be unique within the domain. Each domain will maintain a host name-to-address mapping table for all hosts local to the domain. [12] There may be duplicate host names in different domains.

Domain Name : For our purposes a domain name will indicate a subnet of the internet. There will be a higher authority responsible for assigning these names to subnets as they request internet connectivity. This higher authority will also maintain a domain name-to-domain address mapping table for use by internet gateways.

The syntax of a "completely specified name" is given in the following fashion :

User Name @ Host Name @ Domain Name

An address is a data structure whose format is recognizable by all members of a domain. [13] The addresses for TCP and TP 4-based networks are illustrated in figure 4-4 and described below (recall we must use network layer envelopes in order to perform all services expected of the transport layer internet). [14], [15], [16], [17]

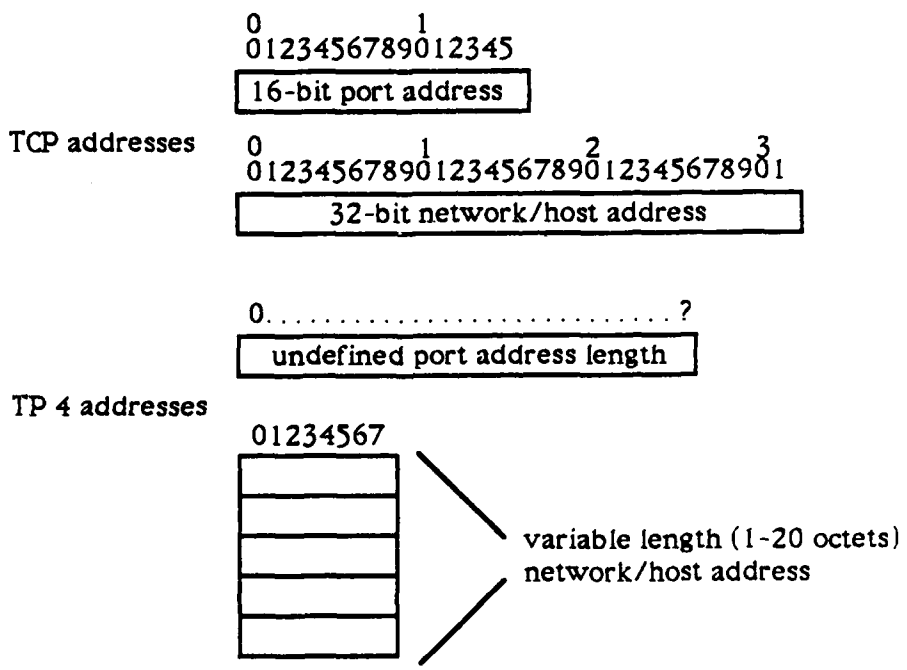


Figure 4-4. TCP and TP 4 ADDRESSES

TCP Addresses : The 16-bit address in the transport envelope is called a "port" and does not have a corresponding name. This address is based upon the particular application specified by the human user of the protocol. For instance, the mail application built on top of TCP has a particular port address, as does the remote login application. [18]

The 32-bit address contained in the network envelope actually consists of two sub-parts. The first is a network address having eight, 16, or 24 bits given to it depending on the number of hosts it expects to accommodate. The remaining bits, 24, 16, or eight, respectively, specify a host within the network.

TP 4 Addresses : These addresses are not as easily defined as in TCP due to the very recent emergence of the standard. Implementations of TP 4 protocols have thus far been for demonstration purposes only, with several vendors gearing up for production versions based upon the demonstrations. [19] The transport envelope does not contain an address, per se, but of the options available to a transport user is the ability to specify the identity of the port requesting service from TP 4. The length of this parameter is not specified in the standard, but a length of 16 bits will be assumed. As for the network envelope's address, it consists of two fields, each of variable length. The first specifies the authority (somewhat analogous to the network address of TCP) responsible for assigning values to the second field. This second field contains host addresses.

With regard to port addresses it is possible to provide uniformity across TCP/TP 4 subnet boundaries. If a host on a TCP subnet wants to send electronic mail to a host on a TP 4 subnet the two must practice the same mail protocol. Otherwise the information, although able to traverse different transport layers, would not make any sense at the destination. If the same upper layer protocols must be used, it makes sense to also assign identical addresses to the ports, whether part of a TCP or TP 4 protocol suite, through which access to the transport layer is granted (e.g. electronic mail would always use port 25, remote login, 21). The (inter)networking industry refers to the values of these access points as being "well-known", with their assignment managed by the internet authority. [20] Each host in the internet is made aware of well-known port addresses. Such is not the case with addresses found in network layer envelopes.

To bridge the gap between network layer address spaces the internet authority registers **internet-unique values** (internet values) for hosts on all connected subnets. These values will play an important part in special "remote name servers" within each subnet. As upper layer interactions are received by transport entities they are parsed into distinct pieces, for example the command

"Mail user_a @ host_a @ domain_a" would consist of four parts, the application (Mail) and the three elements of a completely specified name as described above. Referring to figure 4-5, the steps encountered in processing such a command are illustrated.

Local Host Name-to-Address Table

Host Name
Byu_cs

Host Name	Local Address
Byu_admin	10
Byu_cad	15
Byu_eng	20
Byu_cs	25
Byu_stats	30
Byu_math	35

(Address Returned from Local Table)

Local Host Name-to-Address Table

Host Name
Cs_dept

Subnet Name
Ucb

Host Name	Local Address
Byu_admin	10
Byu_cad	15
Byu_eng	20
Byu_cs	25
Byu_stats	30
Byu_math	35

(Not found in Local Table)

Remote Name Server

Host Name	Subnet Name	Internet Value
Cs_dept	Mit	100
EE_dept	Mit	101
Cs_dept	Ucb	200
EE_dept	Ucb	201
Cs_dept	Ucla	300
EE_dept	Ucla	301

(Value Returned from Remote Server)

Figure 4-5. HOST NAME LOOKUP PROCEDURES

Using its knowledge of well-known ports the entity would fill the port address field of the transport envelope. The transport entity then sends the host name to the host name-to-address table for its subnet. If the name is found in this table a subnet-specific address is returned, otherwise, the remote name server is invoked, again using the parsed host name. [21] In addition to the host name,

the name server must also be supplied with the subnet name of the remote host since it is possible for different subnets to use the same host name. Using this information, each subnet's remote name server contains mappings from all non-local host names to internet values assigned by the internet authority. If the remote name server finds the host name sent to it, the internet value will be returned, if not, the host does not exist in the internet.

As internet values are returned to transport entities they are placed in the address fields reserved for network and host addresses and the message is sent to a meta-protocol gateway connected to the subnet. As the gateway receives the message it looks at the address field containing the internet value and is able to recognize it as such. Available only to gateways is another server, this one responsible for mapping internet values to subnet names and subnet-specific addresses, as illustrated in figure 4-6. Using this server the gateway is able to determine whether the message is bound for a directly connected subnet (every gateway knows the names of subnets attached to it) or if it needs additional forwarding.

In the case of the message being addressed to a directly connected subnet, the subnet-specific address provided by the special gateway server mapping is placed in the network layer envelope's address field and the message is delivered to the directly connected subnet. When the decision to forward the message is made, the internet value is left in tact and the message is sent to another gateway through which the ultimate destination is reachable. The process just described continues until a directly connected subnet is found. Reachability information used to forward messages between gateways would be found in routing tables maintained through a special protocol such as the exterior gateway protocol

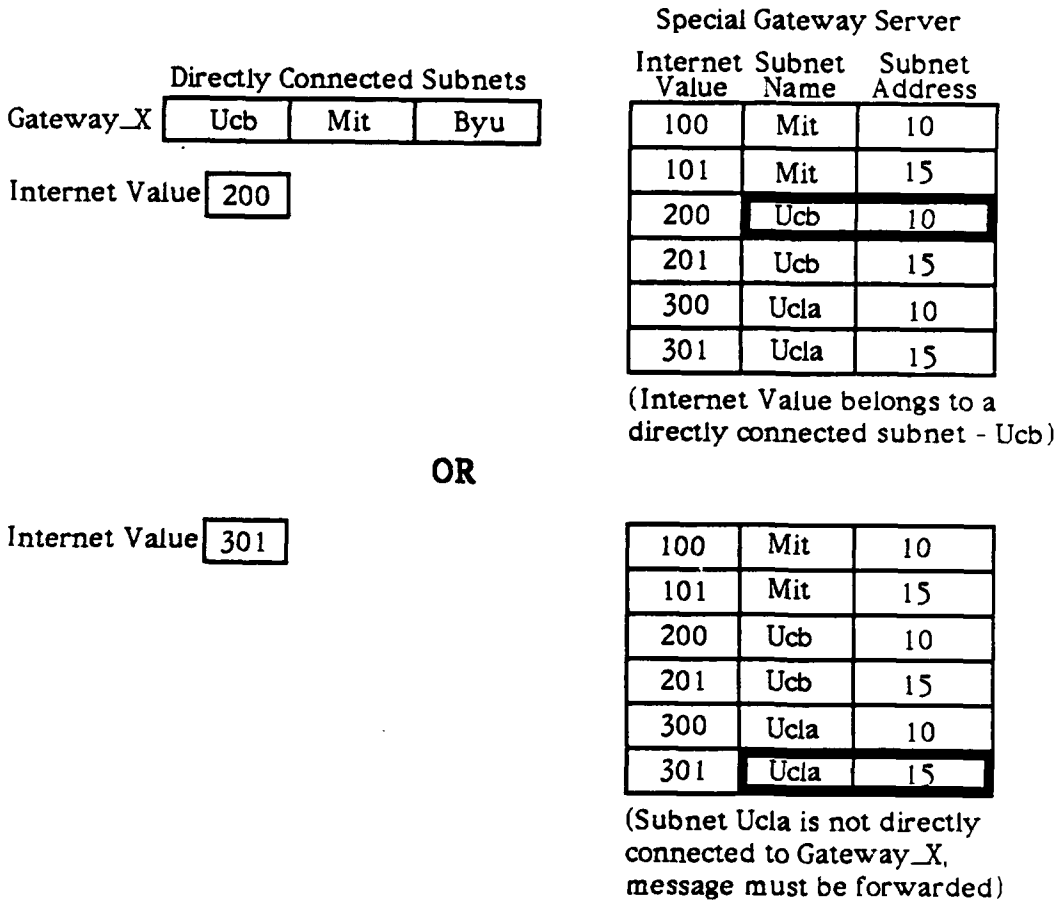


Figure 4-6. GATEWAY INTERNET VALUE LOOKUP PROCEDURES

(EGP) of ARPANET. [22] The details of such a protocol are not covered in this thesis, but the presence of accurate routing table information is assumed.

For this particular name-to-address resolution technique a key design decision comes in determining how large the internet value may be. These values must be representable in the address fields of all connected subnets; therefore the subnet with the smallest address space becomes the limiting factor for internet value size. TCP-based network addresses are 32 bits long. The minimum length of network addresses under TP 4 is not specified, but from [23] it is safe to assume it will be at least as long as its TCP counterpart. Therefore, the internet

value for this thesis will occupy four octets (32 bits). One final issue must be discussed with regard to name-address resolution.

A common way to view each layer within an OSI-based protocol suite it to decompose each into sub-layers (just as has been done with the meta-protocol). [24] If transport layers (TCP or TP 4) are treated as shown in figure 4-7, the provision of services for messages coming from upper layer protocols in the same host and those coming from meta-protocol gateways is made clearer.

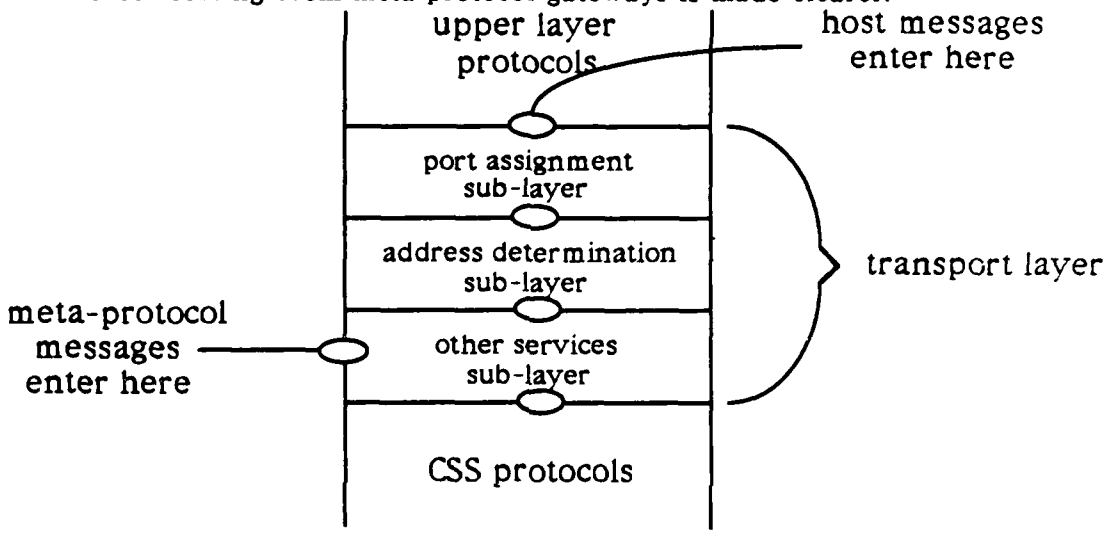


Figure 4-7. TRANSPORT PROTOCOL SUB-LAYERS

The "port assignment" sub-layer is charged with furnishing values for the port address field in the transport layer envelope. The "address determination" sub-layer obtains network envelope addresses using mapping functions like those described in this section and passes them on to the network layer. Messages coming from upper layer protocols pass through each transport sub-layer before being passed on to the network layer, whereas gateway-produced messages already having their port address filled and in possession of the appropriate network layer address, need only enter at the "other services" sub-layer.

When more than one subnet is attached to a gateway connectivity information alone (number of intermediate hops) may not indicate a superior

choice. As a further qualification, gateways should try to match optional performance parameters of outgoing messages against those available in directly connected subnets. Such options inform the underlying CSS that it should make every effort to satisfy the performance demands conveyed by these parameters. Otherwise, the unreliable CSS is free to do as it pleases with the message.

4.3 Optional Parameter Preservation Sub-layer

This sub-layer takes its theoretical justification from the ideas of Redell and White [25], while the suggested implementation is based upon the work of Gelotte [26]. Redell and White suggest two approaches to problems such as this. Both are formed by creating a set of options, determined by analyzing all that are currently available. This is not an unreasonable approach since transport and network options are fairly well defined and limited in number. The first approach, or **least common denominator**, would restrict the set of available options to that of the smallest subnet projected to become part of the internet. While this does allow even the most basic of subnets to be incorporated smoothly, it severely stifles the potential of more sophisticated subnets. The other alternative, or **universal superset**, makes it possible for any conceivable subnet to have all of its options represented by incorporating every possible option in the set. The meta-protocol will use the latter technique since it allows even the most sophisticated subnet options to be made available to any other subnet capable of handling them, while at the same time forcing a relatively small amount of overhead on the gateways providing this function. To represent the superset of options an "options vector" is chosen with an element in the vector corresponding to a selectable option. Each gateway maintains a vector for each subnet it is connected to (outgoing vector). The information required to populate each outgoing vector is supplied through special messages sent by

subnets addressed specifically to the gateway. This is possible since gateways are treated as any other host on the internet. [27] The presence of an option is depicted by a one (1) in that element of the vector. As part of each subnet's message conversion to the meta-format, another option vector is produced. This one (incoming vector) represents those options requested by the converted message, again the presence of a one (1) signifying a selected option. Internal to the gateway, the incoming vector is treated as an array with "n" rows and one (1) column, whereas outgoing vectors are arrays with one (1) row and "n" columns. By performing a matrix multiplication operation on each pair of incoming-outgoing vectors and concerning ourselves with only the diagonal elements of the resulting "n" by "n" matrix, as illustrated in figure 4-8, a measure

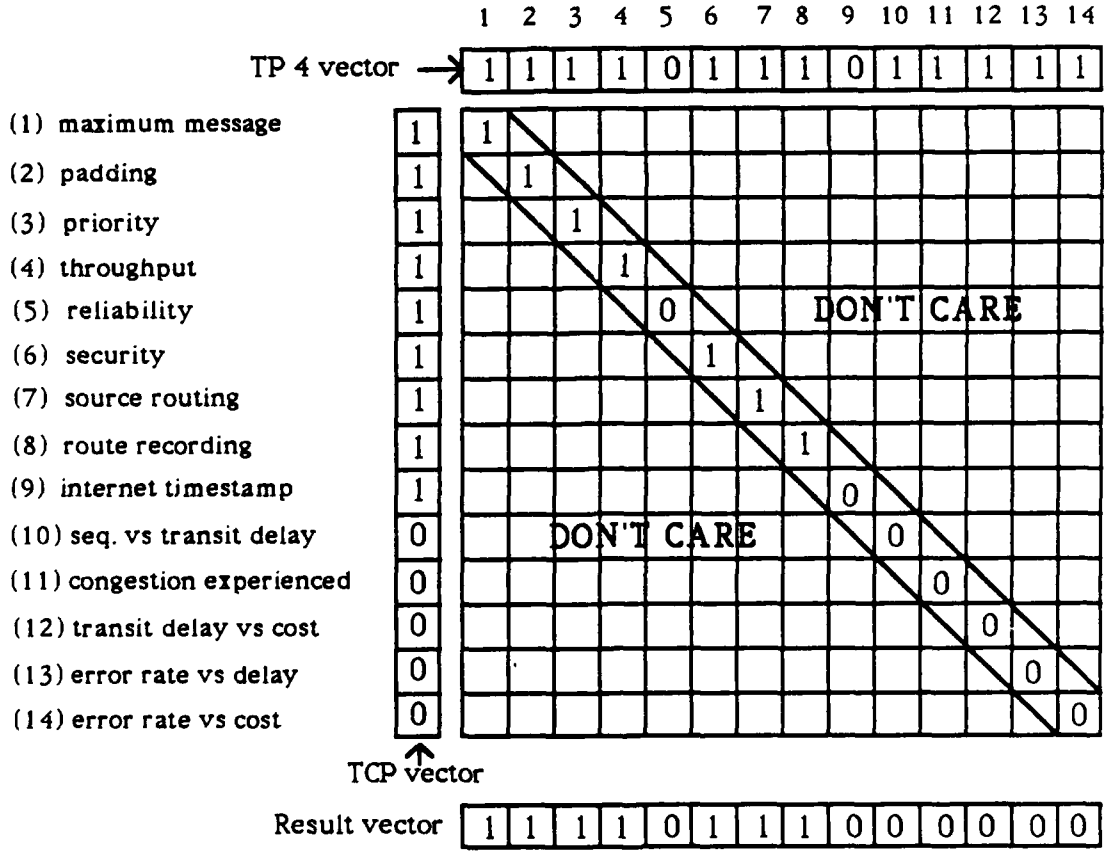


Figure 4-8. OPTION VECTOR MULTIPLICATION

is obtained as to the relative ability of each connected subnet to support the requested options. This measure is then used to assist the gateway in making its decision as to which outgoing subnet should be used to forward the message to the ultimate destination. The combination of information from this sub-layer, the name-to-address sub-layer, and the assumed routing table prepares a meta-protocol gateway to make an intelligent decision as which subnet should next perform those functions traditionally associated with a communications protocol suite.

Although not in proper sequential order (see figure 4-3), the fifth sub-layer of the architecture will next be explained due to the reliance of sub-layers three and four upon its functions.

4.4 Traditional Services Sub-layer

By using the connection-oriented nature of each subnet as exists in independent TCP/TP 4 networks, the meta-protocol is able to disregard those problems of chapter three falling under the "Traditional Transport Services" section. As the entrance half of the meta-protocol gateway performs its conversion function, it simulates an upper layer request to the transport layer. The traditional connection management and transparent delivery mechanisms process the simulated request as any other request. Thus, the traditional services (PAR, sequencing, checksum) are exercised by each subnet and the architecture has, by default, bridged part of the heterogeneity gap existing among protocol suites. In this way the splicing effect of actual connections into a single virtual connection is realized. Returning to the logical flow as depicted figure 4-3, sub-layer three will now be explained.

4.5 Fragmentation Sub-layer

In response to the problem of different subnet message sizes there are two general techniques in use -- intranet and internet fragmentation. [28]

4.5.1 Internet Fragmentation

This approach results in any fragmentation performed at gateways being propagated throughout the internet. Messages determined too large for the selected forwarding subnet are fragmented into smaller ones. These "derived messages" are then treated independently by other internet hosts and gateways. Derived messages may themselves be fragmented at other gateways. Only at the ultimate destination host are all fragments reconstituted into the original message. [29] Although a popular technique, the meta-protocol's reliance upon subnets to create reliable transport connections, each dealing with fragmentation/reassembly of messages, prohibits it use.

4.5.2 Intranet Fragmentation

Intranet fragmentation, breaking up of messages at the entrance gateway with reassembly performed by either an exit gateway or destination host, is the approach to be used in a meta-protocol internet. [30] Using intranet fragmentation resolves meta-protocol gateways of any direct fragmenting responsibilities. Exit halves do become indirectly involved in this process by advising entrance halves as to the size of the upper layer data it will be passing to it. Both TCP and TP 4-based networks contain fields in their envelopes, as illustrated in table 4-1, making this function relatively straight-forward. Two scenarios, TCP to TP 4 and TP 4 to TCP, are possible.

Table 4-1. ENVELOPE FIELDS USED FOR FRAGMENTATION

TCP Length Fields	
"total length"	- total length of transport and network envelopes, plus upper layer data
"ihl"	- internet header length, or length of the network envelopes measured in 32-bit words
"data offset"	- length of transport envelope in 32-bit words
"identification"	- identifier used to group fragmented messages together
TP 4 Length Fields	
"segment length"	- total length of transport and network envelopes, plus upper layer data
"LI" (network)	- length of network envelope in octets
"LI" (transport)	- length of transport envelope in octets
"data unit identifier "	- identifier used to group fragmented messages together

4.5.2.1 TCP to TP 4 Subnets

For each TCP message the exit half takes the "total length" field (measured in octets) of the network envelope and subtracts the "ihl" field (multiplied by four since it is measured in 32-bit words). The result of this operation gives the length (in octets) of the transport envelope and upper layer data. The "data offset" (also multiplied by four) is then subtracted from this intermediate length. When messages are actually fragments of a larger message, indicated by an identical value in the network envelope's "identification" field, the result of the latter subtraction for every fragment would also be added together. This summation, or the latter subtraction when no fragmentation was encountered, gives the exit half the value (in octets) it should relay to the entrance half.

4.5.2.2 TP 4 to TCP Subnets

By making the following field name substitutions, logic similar to that as used in the TCP-TP 4 scenario communicates TP 4 message lengths to TCP entrance halves.

Use this TP 4 field	For this TCP field
"segment length"	"total length"
"LI" (network envelope)	"ihl"
"LI" (transport envelope)	"data offset"
"data unit identifier"	"identification"

Only one difference from the previous scenario exists -- the two TP 4 "LI" fields are already measured in octets, therefore, no multiplication is required.

. Utilization of reliable subnet hops also simplifies the last sub-layer definition.

4.6 Sequence Number Preservation Sub-layer

To cope with the differences in sequencing strategies between TCP and TP 4-based networks, the meta-protocol architecture chooses an extremely effective technique -- it doesn't do **anything**. Every potential mishap, with regard to message sequencing and delivery by a Class C CSS, is overcome through the reliable nature of both TCP and TP 4 protocols. Out-of-order messages are resequenced, duplicates are ignored, and fragments are reassembled; all transparent to the gateway. When receiving a message in need of forwarding, an exit half only extracts those envelope fields needed to satisfy the functions described previously in this chapter. All other information, including the sequence number, is stripped from the message. What remains from this "envelope opening" process is the unmodified upper layer data inserted by the original internet source. Such message stripping must be performed if the gateway is to properly simulate an upper layer interaction with the subsequent subnet (see "Traditional Services Sub-layer"). This unenveloped message is then passed on to the selected entrance half

where subnet-specific sequence number allocation techniques are used. As viewed from the internet, a single message has as many different sequence numbers as number of subnets it traverses before reaching its final destination.

4.7 Summary

In defense of the chronology implied by figure 4-3 for accomplishing required internet functions, the following rational was used. Before any decision as to forwarding of messages may be reached, the address of the named host must be obtained. The message's originating subnet is the first to become involved in this task as it attempts to determine whether the message is bound for a local or remote destination. When remote messages are transmitted, intermediate gateway halves explicitly relay internet values from subnet to subnet until the destination host is reached. The assumed presence of routing table information at each gateway provides knowledge of which subnets are candidates for message forwarding, based exclusively on connectedness to the final location. In hopes of narrowing the number of possible forwarding subnets, the meta-protocol matches each candidate's option vector against the vector belonging to the message. Fragmentation/reassembly functionality is only indirectly provided by the meta-protocol. Message lengths are relayed, but subnet-specific protocol suites ensure messages meet the size requirements levied by the subnet. As for the remaining sub-layers, gateway halves simply unwrap/wrap network and transport envelopes, giving each subnet the illusion that the upper layer data has originated from one of its hosts. At this point, sequencing and all other services normally associated with the reliable, error-free delivery of information are of no concern to the meta-protocol. They are supplied in accordance with local transport protocol techniques.

Though only a subset of the problems found in connecting heterogeneous transport protocols have been addressed, benefits of using layer four in support of a meta-protocol architecture have been demonstrated. The question that might next be asked is, "What about other internetworking strategies, wouldn't they work?"

CHAPTER 5

Alternative Internetworking Strategies

A positive response for the previous question is appropriate for each of several alternative internetworking strategies. However, a question of more importance is, "How do these differ from the architecture of this thesis?" CCITT's answer to the internetworking problem, X.75, has already been discussed and its ability to interconnect only X.25 networks given as grounds for meta-protocol superiority. Three other designs have received attention in internetworking literature, with the internet protocol approach being used by several vendors for experimental and operational internet implementations. Each will be dealt with in the following sections.

5.1 Global Internetwork Standard

Without a doubt, this is the best approach to connecting (dis)similar networks. The only problem is getting the broad collection of networking organizations to adopt such a standard. As stated in chapter one, networks are designed for a particular applications at the request of, most likely, a single organization. As these designs are implemented other organizations have the opportunity to accept or reject it. The opposite is true for an international standard. Those tasked with defining such a standard are not concerned with the needs of individual organizations. They are able to include and/or exclude features without approval from the potential users (those creating the standard are the users). ISO, aware of the situation, has proposed a global internetworking standard, Draft International Standard (DIS) 8473, "Protocol for Providing the Connectionless-mode Network Service". [1] It resides at layer three

of the OSI model and once receiving International Standard (IS) status adherence to it by a large portion of the internetworking community is expected.

5.2 Internet Protocol

There have been several experimental internets created using this technique and the largest operational internet in existence, the DOD's ARPANET, also chose this approach. With regard to the OSI model, an internet protocol (IP) is located between the network and transport layers (see figure 5-1). It provides global addressing and routing at gateways much like the meta-protocol. [2] IP information is treated as another envelope wrapped around incoming messages and stripped off at the final destination.

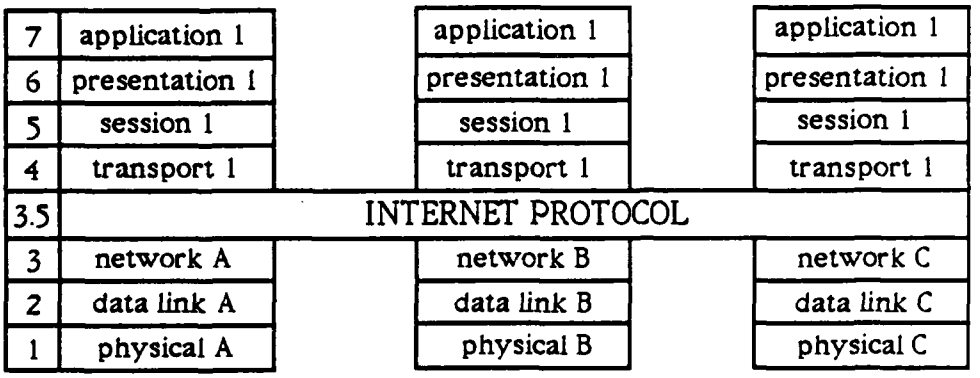


Figure 5-1. COMMON INTERNET PROTOCOL

The drawback to this approach comes from its insistence that every subnet host and internet gateway implement the IP in addition to its conventional protocol layers. [3] With each protocol layer comes more processing overhead and proportionately longer transmission delays. The message carrying capacity of the underlying CSS remains the same whether an IP layer is used or not. However, due to the additional IP envelope, the net throughput (ratio of application-specific data to control information carried in envelopes) is lowered. As for a meta-

protocol-based solution, interconnectivity issues are handled transparent to subnet users. Only gateways need implement the meta-protocol.

5.3 Pure Protocol Translator

Although the architecture of this thesis involves the conversion of protocols, it only requires two such operations on the part of each connected subnet -- one to the meta-protocol format and another from the meta format to a subnet-specific protocol. For a "pure" protocol translation-based internet, the number of conversions required (see figure 5-2) in a worst case scenario turns out on the order of "N squared", with N being the number uniquely defined subnets. When only a small number of heterogeneous subnets exists this may not present such a great problem, but when dealing with larger numbers the amount of special-purpose translation software becomes unmanageable.

With N = 5, $N^2 - N$, or 20 Translations Required (Unidirectional Arcs)

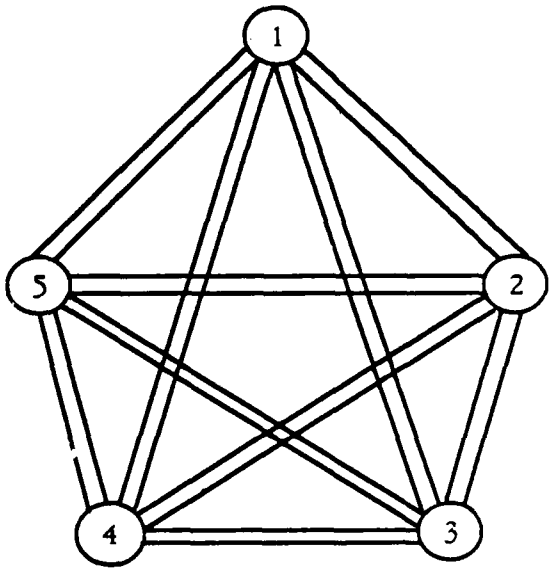


Figure 5-2. PURE PROTOCOL TRANSLATOR

Each subnet in a meta-protocol internet is responsible for two conversions, thus reducing the total to "2 times N" -- a significant difference from the pure

translations option when dealing with relatively large Ns. Just as discussed with respect to preservation of optional parameters in the meta-protocol architecture, situations will arise using a pure translator where regardless of the sophistication of the translation, services of one subnet cannot be matched by another. [4] The last problem with this approach is found in the placement of the translator relative to other protocol layers. CSS layer standards are firmly established in today's networking community, in fact, these layers have been "chipped" using VLSI technology. [5] Such standardization would allow for a relatively straightforward translation implementation. However, when considering higher protocol layers, the unique needs of different user groups and the lack of agreement on what should be included in these layers would make a pure translation approach extremely difficult. [6]

Having discussed its relative strengths and weaknesses, the next chapter describes the effectiveness found in using a simulated meta-protocol internet.

CHAPTER 6

Meta-Protocol Simulation

To demonstrate the feasibility of using the meta-protocol approach for connecting TCP and TCP 4-based subnets a simulation has been implemented using Turbo Pascal, version 3.0. [1] Although Turbo Pascal does not provide the low level programming capabilities needed to precisely represent all the fields found within TCP and TP 4 envelopes, it proved sufficient for simulation purposes. The internet environment is represented as a collection of linked lists; one with a node for each subnet registered on the internet, one for meta-protocol gateway half pairs, one used to represent the special server available only to gateways for mapping internet values to directly connected subnet addresses, and another containing the names and addresses of "well-known" ports. Each node of the subnet list contains three other linked lists; the first is a list of hosts local to the subnet with their local addresses, the second, a list of all hosts remotely located along with their internet values, and the third, a list of all gateway halves connected to the subnet. The latter three lists provide the means for any host to determine the appropriate address values for the hosts it wants to communicate with and to know which gateways are available for remote traffic forwarding. Consistent with the assumption of routing table information at each gateway, the simulation does not have an implicit router. Rather, the simulation provides a list of gateways and subnets available at each intermediate stop and the operator makes the appropriate choice. Only when remote messages reach an entrance gateway half with the final destination subnet directly connected, does the simulation make any routing decision. This it does by immediately delivering the message to the directly connected destination.

With regard to the relative difficulty of realizing different aspects of meta-protocol functionality, a ironic situation was discovered. By far, name-to-address resolution required the greatest amount of attention, while that of option preservation was implemented using a relatively simple technique. The irony comes from the measure of subnet compatibility these two functions give the user of the simulation. As will be demonstrated in several simulation scenarios, addressing differences among heterogeneous hosts were easily overcome due the naming convention adopted. However, the simple method for displaying the message's option vector at each intermediate stop of its internet path reveals how the performance intentions of the originating host are susceptible to deterioration. This failure to match optional performance parameters is not a deficiency in the meta-protocol architecture, rather, it results from different implementations of the same abstract concept -- reliable, error-free delivery of application-specific information. Regardless of the technique employed, certain features of one transport protocol will not translate to a different protocol.

The specific options selected for use in the simulation are not to be interpreted as the only ones of importance. They were chosen so as to include some found only in TCP and others only in TP 4, while others are common to both protocols. The intent is to demonstrate the **feasibility** of using a meta-protocol approach for matching these parameters, not to produce an **operational** internet. Although not portrayed by the option preservation technique, the following point should be remembered when discussing each of the five scenarios. The fact that both TCP and TP 4 have parameters of the same name (e.g. security, throughput, priority) **does not** imply a compatible translation. For example, the TCP priority parameter may take on one of eight values represented by three bits, whereas TP 4 has reserved two octets for its priority option. [2],

[3] For situations such as this a more detailed correlation function must be used to retain the characteristics of the internet message as it passes from subnet to subnet.

After experimenting with the simulation, it was determined that all possible subnet traversal combinations and message convolutions could be represented using five source-destination scenarios. Each scenario begins with a specification of the source and destination of the message, the upper layer application (e.g. mail, ftp), and the message's optional parameters. The source **must** be entered as a completely specified name (user, host, network). The destination consists of at least an application name (mapped to a well-known port) and a user name. If either the host or network name is omitted the corresponding name from the source is used. Whenever a remote destination is specified, the simulation provides a set of three message "snapshots" (see figure 6-1) for each subnet it must pass through. The first snapshot shows the message as it appears just before entering the exit gateway half, the second, in the meta-protocol format, and the third immediately after leaving an entrance gateway half. The contents of these snapshots vary according to type of protocol suite encountered.

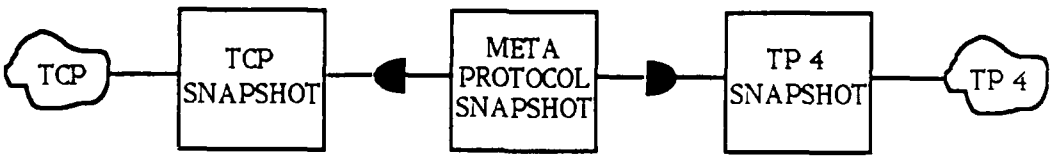


Figure 6-1. INCREMENTAL MESSAGE REPRESENTATION

Only those envelope fields needed for conversion to/from the meta-protocol format are presented in the simulation's snapshots. Consistent with the explanation chapter four of traditional transport services; sequence numbers, flow

control windows, etc. are of concern only to each individual subnet. Obviously, complete envelopes would exist in an operational internet.

One of the fields found in each snapshot is the address of the destination network/host. This address may appear as a subnet-unique value or as an internet value. TCP network/host addresses were previously shown as being 32 bits long, while the TP 4 counterpart was presented as a variable length field. For simulation purposes only, a six octet value will be used for TP 4 network/host addresses. The distinguishing feature between subnet-unique and internet addresses in the simulation is found in the most significant bit. Subnet addresses will always have a zero (0) value in this position, while internet addresses set the bit to a one (1). This addressing technique and other features of a simulated meta-protocol internet will be explained in the following sections.

6.1 Simulated Internet Topology

To support the five message exchange scenarios an internet consisting of six (6) subnets, three TCP and three TP 4, and five (5) gateways (see figure 6-2) was constructed.

6.2 Message Exchange Scenarios

To demonstrate the full range of meta-protocol operation, five messages, one corresponding to each of the following categories, were created and submitted to the simulation :

1 - Source and destination on same subnet
2 - Source and destination on adjacent, homogeneous subnets; adjacent defined as being connected by the same gateway
3 - Source and destination on adjacent, heterogeneous subnets
4 - Source and destination on non-adjacent, homogeneous subnets. Two sub-scenarios must be considered in this situation :
    - homogeneous (with respect to source and destination) intervening subnets
    - heterogeneous intervening subnets
5 - Source and destination on non-adjacent, heterogeneous subnets.

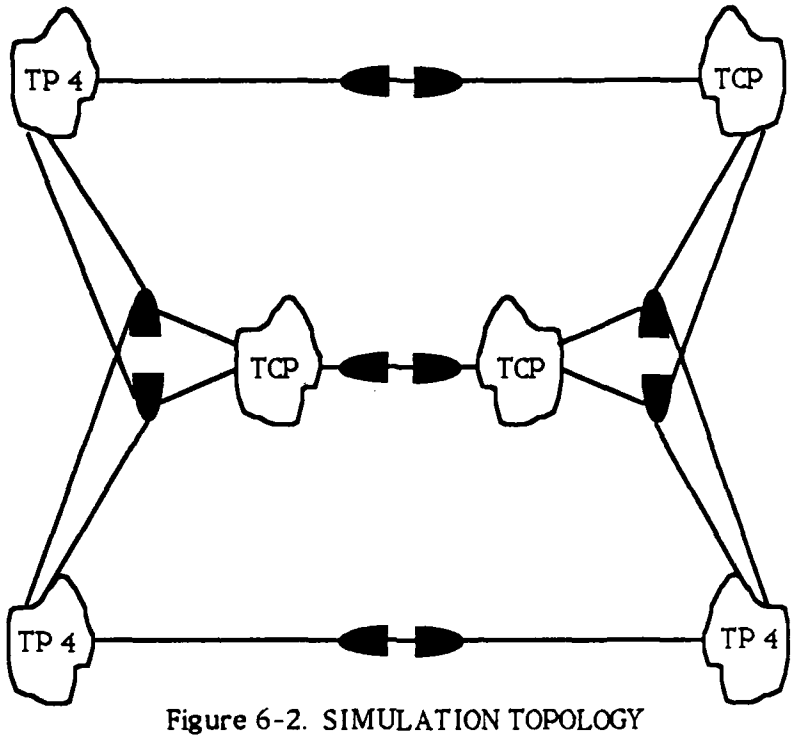


Figure 6-2. SIMULATION TOPOLOGY

6.2.1 Same Subnet

This scenario is trivial as far as meta-protocol functionality is concerned. All processing is handled by the local protocol suite. All options requested at the source are preserved and used for message delivery since the destination practices the same protocol.

6.2.2 Adjacent, Homogeneous Subnets

Although destined for a remote host, thus requiring meta-protocol services, this scenario has the same results as the previous one. Any options specified by the source must come from the same set of options as available at the destination, thus they are kept in tact when crossing the subnet boundary. In this case, the meta-protocol performs much like CCITT's X.75 internetworking solution.

6.2.3 Adjacent, Heterogeneous Subnets

When faced with this scenario, the full capabilities of the meta-protocol are exercised. For the first time, destination address information must be conveyed in formats differing in both size (32-bit vs 48-bit) and semantics (TCP and TP 4 allocate different portions of the overall address space for designating network and host). As for optional performance parameters selected by the source host that are actually used for delivery by the destination protocol, figure 6-3 illustrates the possible combinations. Only when all parameters chosen by the source are available in both TCP and TP 4 (figure 6-3a) would there be a one-to-one mapping. Using a worse case scenario, the source may only select parameters used on its local subnet (figure 6-3b). Upon arrival at the heterogeneous subnet, a message completely stripped of performance parameters will be processed. The upper layer data, in this a case, would be subject to the minimum performance features offered by the CSS. Any number of possibilities exist between these two extremes (figure 6-3c).

6.2.4 Non-adjacent, Homogeneous Subnets

Considering the first sub-scenario (homogeneous intervening subnets), it reduces to that found in scenario two. The only difference being the relaying of homogeneous protocol information through more than one gateway.

The second sub-scenario also reduces to one already described -- scenario three, again with one difference : the homogeneous destination subnet is now capable of receiving a message containing a degraded set of performance parameters.

Options Available	TCP	TP 4
padding	X	X
security	X	X
source routing	X	X
route recording	X	X
priority	X	X
stream id	X	
timestamp	X	
throughput	X	X
error rate	X	X
acknowledgement time		X
delay		X
sequencing vs delay		X
delay vs cost		X
error rate vs delay		X
error rate vs cost		X

(a)

TCP Source - options selected	
padding	X
security	X
source routing	X
route recording	X
priority	
stream id	
timestamp	
throughput	X
error rate	X
acknowledgement time	
delay	
sequencing vs delay	
delay vs cost	
error rate vs delay	
error rate vs cost	

Subnet Boundary

TP 4 Destination - options preserved	
padding	X
security	X
source routing	X
route recording	X
priority	
stream id	
timestamp	
throughput	X
error rate	X
acknowledgement time	
delay	
sequencing vs delay	
delay vs cost	
error rate vs delay	
error rate vs cost	

(b)

TP 4 Source - options selected	
padding	
security	
source routing	
route recording	
priority	
stream id	
timestamp	
throughput	
error rate	
acknowledgement time	X
delay	X
sequencing vs delay	X
delay vs cost	X
error rate vs delay	X
error rate vs cost	X

Subnet Boundary

TCP Destination - options preserved	
padding	
security	
source routing	
route recording	
priority	
stream id	
timestamp	
throughput	
error rate	
acknowledgement time	
delay	
sequencing vs delay	
delay vs cost	
error rate vs delay	
error rate vs cost	

(c)

TP 4 Source - options selected	
padding	X
security	X
source routing	X
route recording	
priority	X
stream id	
timestamp	
throughput	
error rate	X
acknowledgement time	
delay	
sequencing vs delay	X
delay vs cost	X
error rate vs delay	
error rate vs cost	X

Subnet Boundary

TCP Destination - options preserved	
padding	X
security	X
source routing	X
route recording	
priority	X
stream id	
timestamp	
throughput	
error rate	X
acknowledgement time	
delay	
sequencing vs delay	
delay vs cost	
error rate vs delay	
error rate vs cost	

Figure 6-3. OPTION EXCHANGE SCENARIOS

6.2.5 Non-adjacent, Heterogeneous Subnets

We could use the same two sub-scenarios from the previous section for this discussion, but both would eventually reduce to the same situation as found in scenario three and figure 6-3. Even if every intervening subnet was of the same type as the source, the final gateway would be faced with transferring the message to a heterogeneous (with respect to the source) destination. Accompanying this heterogeneity would be the same potential for message degradation as explained in scenario three.

6.3 Summary

The preceding sections should reveal the goal of each message sent from a host on a meta-protocol-based internet : never go through a subnet exercising a different transport protocol than that of the source. Although this would solve the problem of loosing performance parameters, it is not a realistic. As stated in scenario five, even if every subnet in the message's path were of a homogeneous nature, the message may be addressed to a host on subnet using the other protocol. Obviously, this subnet cannot be avoided.

CHAPTER 7

Summary and Conclusions

7.1 Summary

An architecture designed to overcome the functional differences between the DOD's de facto layer four protocol standard, TCP, and the TP 4 transport protocol developed by ISO, soon to become an international standard, has been proposed. Attention has been given to only these protocols due the personal interest of the author in the (inter)networking efforts of the DOD. Certainly, other protocols could be considered as members of a meta-protocol internet. There are many reasons, economic and technical in nature, for trying to connect previously independent networks into a cooperating internet of computing resources. Using the OSI model, as proposed by ISO, in determining the location for internetworking functionality leads to a choice among its seven layers. Of the seven, layer four is the only one concerned with providing a reliable, error-free stream of information to the upper, application-specific layers. For this reason, the meta-protocol was designed to "splice" together physical transport connections from individual subnets into a single logical connection between source and destination hosts. This splicing effect is supplied by internet gateways and in so doing, allows the architecture to provide traditional transport layer services by default. Each physical transport connection operates as if it was independent of any other connection. At gateway halves, the meta-protocol extracts and inserts the information it needs to perform its interconnecting role. The latter information includes internet values assigned by a single allocating authority that bridge the addressing dissimilarities between TCP and TP 4, and optional performance parameters carried by internet messages. By way of a meta-protocol simulation, the concepts proposed in this thesis were given credibility.

It was found that any host, regardless of location, could be addressed by any other host on the internet. The simulation also demonstrated performance parameters as posing the greatest threat for maintaining the integrity of messages when crossing heterogeneous subnet boundaries.

7.2 Conclusions

The realization of a problem usually bring about a concerted effort avoid those practices contributing to it. This has not been the case with regard to connecting computer networks. Especially in the Local Area Network (LAN) arena, the opposite has occurred. LAN vendors continue to establish proprietary higher layer protocols (transport layer and above) due to the specialized tasks performed by their customers. From an economic standpoint, they are justified in doing so : protocols tailored to specific needs of LAN users allow for increased productivity. [1] Although ISO is making efforts to overcome such diversity by designing a family of International Standard communication protocols, the need for protocol conversion techniques will grow in the future.

The research performed for this thesis has indicated the ability to provide connectivity between heterogeneous networks (LANs or otherwise), but prior agreement among participants must be reached before an attempt is made. The meta-protocol architecture supplies connectivity among heterogeneous networks by combining the strengths found in other internetworking techniques. End-to-end reliability, as found in an Internet Protocol, is provided by a collection of intermediate, reliable connections. Special-purpose translation software, as compared to a pure protocol translator, is reduced. Taken together, these benefits indicate operational meta-protocol-based internets would meet the need for expanded connectivity among heterogeneous computer networks.

7.3 Areas for Further Research

The natural predecessor to a proposed architecture is an operational implementation. An internet created in accordance with the meta-protocol architecture would find a large portion of the functionality already in place -- the individual network protocol suites used to populate the internet. A practical name-to-address resolution technique and the use of gateway protocols for carrying option preservation information, point out areas requiring additional attention. Finally, a method for adding other transport protocols (other than TCP and TP 4) to a meta-protocol internet might be considered.

7.3.1 Name-to-Address Resolution

The technique employed by simulation for distinguishing subnet-specific and internet addresses (high order bit status) would not be a practical solution for actual TCP and TP 4 subnets. They already have semantics associated with this and other bits in their address fields. A true address resolution technique would have to involve members from each subnet desiring connection to the internet. Such a forum would allow agreement as to the structure of addresses reserved for internet usage. An internet authority would have to be established, with registration of subsequent subnets managed by this authority.

7.3.2 Option Preservation

For alerting gateways of the option carrying capability of subnets, the routing table protocols assumed for this thesis provide the ideal vehicle. Presently, these protocols include mechanisms for determining distances between gateways, error conditions discovered at gateways, and the status of subnets attached to the internet. A data structure containing a subnet identifier and those performance parameters present in the subnet could also be incorporated.

At specified intervals, subnets would send "option update" messages to each of its connected gateways. Gateways would then have access to the most current information when pairing incoming messages with forwarding subnets.

7.3.3 Additional Transport Protocols

The existence of a well-defined meta-protocol format suggests a possible short-cut solution for subnets in writing their conversion routines to/from the meta-protocol -- a "conversion routine generator". Such a generator, based upon those used in the construction of compilers, [2] would accept a "specification language" describing the meta-protocol conversion routines as input and produce the source code for the routines. The use of source code generators eliminate the chance of erroneous conversion routine software. Once the generator has been written and its output deemed reliable, subsequent conversion routines are guaranteed to operate correctly. The specification language for the meta-protocol might include the means for indicating address sizes, maximum message length, and optional performance parameters used by the subnet.

The variety of transport protocols used in today's computer networks were designed for specific reasons. As connectivity now becomes more important than those original reasons, researchers are looking for promising solutions to the problem. It is hoped that this thesis will provide the motivation for considering an implementation based up the meta-protocol architecture.

END NOTES
CHAPTER 1

[1] GROE86, p. 288
[2] STAL84, p. 215
[3] WITT86, p. 3
[4] ROBE70, p. 543
[5] PISC86, pp. 120-121
[6] KAHN72, p. 1397
[7] ROBE74, P. 46
[8] Discussion with Dr. Robert Linebarger, February 1987
[9] STAL85, p. 437
[10] PISC86, p. 135
[11] MARI66, p. 426
[12] Course Lecture from Mr. Dana Doggett of Novell, Inc., June 1987
[13] Course Lecture from Mr. Thomas Bogart of Novell, Inc., June 1987
[14] HURS84, p. 139
[15] HURS84, p. 140
[16] TANE81, p. 356
[17] STAL85, p. 438
[18] HIND83, p. 42
[19] CERF78, p. 1
[20] STAL87, p. 339
[21] WEIS87, p. 141
[22] CALL83, p. 1388
[23] BOOC87, pp. 33, 34
[24] ISO84, general reference
[25] CALL83, p. 1389
[26] SHEL82, p. 112
[27] SUNS80, p. 149
[28] SUNS77, p. 185
[29] CALL83, p. 1389

END NOTES
CHAPTER 2

[1] GEE80, p. 14
[2] SUNS81, p. 345
[3] GROE86, p.289
[4] SUNS80, p. 147
[5] SCHN83, p. 17
[6] SUNS77, p. 175
[7] GEE80, p. 23
[8] KNIG83, p. 1395
[9] ISO86, p. 5
[10] STAL85B, p. 106
[11] ISO87, p. 9
[12] KNIG83, p. 1394
[13] ISO86, p. 3
[14] STAL87, pp. 42, 43
[15] ISO86, p. 3
[16] ISO84, p. 9
[17] KNIG83, p. 1394
[18] STAL84, p. 201
[19] Discussion with Dr. Robert Linebarger, June 1987

END NOTES
CHAPTER 3

[1] TANE81, p. 325
[2] STAL85B, p. 101
[3] SUNS78, p. 442
[4] MIL83, p. 14
[5] MIL83, p. 81
[6] ISO87, p. 19
[7] STAL84, p. 202
[8] ISO84B, p. 66
[9] MIL83, p. 76
[10] ISO87, p. 18
[11] STAL85, pp. 480, 481
[12] SUNS81, p. 355
[13] STAL85, p. 509
[14] CALL83, pp. 1390, 1391
[15] TANE81, p. 364
[16] MIL83, p. 69
[17] ISO84B, p. 50
[18] MIL83, p. 92
[19] MIL83B, p. 32, 35
[20] ISO84B, pp. 122-127, 135
[21] ISO86B, pp. 39-45

END NOTES
CHAPTER 4

[1] POST80, p. 605
[2] CERF78B, pp. 1392, 1393
[3] SHEL82, p. 112
[4] SHOC78, p. 392
[5] XERO81, p. 55
[6] TANE81, p. 336
[7] OPPE81, p. 14
[8] DALA81, p.240
[9] CLAR82, p. 5
[10] OPPE81, chapter 9
[11] SHOC78, p.389
[12] CLAR82, p. 3
[13] SHOC78, p. 389
[14] MIL83, p. 3, 90
[15] MIL83B, p. 25, 26, 31
[16] ISO84B, p. 122
[17] ISO85, chapter 8
[18] Conversation with Mr. Kelly McDonald, October 1987
[19] MCCO87, p. 30
[20] XERO81, pp. 17, 58
[21] CLAR82, p. 3
[22] MILL84, general reference
[23] ISO85, pp. 21, 25
[24] WARE83, p. 1386
[25] REDE83, p. 60
[26] GELO87, chapter 3
[27] SHEL82, p. 119
[28] SHOC79, p. 383
[29] BOGG80, p. 616
[30] SHOC79, p. 384

END NOTES
CHAPTER 5

[1] ISO86B, general reference
[2] MIL83, p. 8
[3] SUNS77, p. 185
[4] STAL85, p. 464
[5] Discussion with Dr. Robert Linebarger, February 1987
[6] Conversation with Mr. Dale Neibaur of Novell, Inc., March 1987

END NOTES
CHAPTER 6

[1] BORL85, general reference
[2] MIL83A, p. 32
[3] ISO84B, p. 126

END NOTES
CHAPTER 7

[1] Discussion with Dr. Robert Linebarger, November 1987
[2] AHO86, pp. 128, 257, 572

CITED WORKS

AHO86 Aho, Alfred V., ^ al, Compilers : Principles, Techniques, and Tools, Addison-Wesley P· ɔlishing Company, 1986

BOGG80 Boggs, David R. et al, "Pup : An Internetwork Architecture", IEEE Transactions on Communications, VOL. COM-28, NO. 4, April 1980, pp. 612-624

BOOC87 Booch, Grady, Software Engineering with Ada, The Benjamin/Cummings Publishing Company, Inc., 1987

BORL85 Borland International, Inc., Turbo Pascal Version 3.0 Reference Manual, 1985

CALL83 Callon, Ross, "Internetwork Protocol", Proceedings of the IEEE, VOL. 71, NO. 12, December 1983, pp. 1388-1393

CERF78 Cerf, Vincent, "The Catenet Model for Internetworking", in DDN Protocol Handbook, Vol. 2, December 1985, pp. 2-7 - 2-25

CERF78B Cerf, Vincent and Kirstein, Peter T., "Issues in Packet-Network Interconnection", Proceedings of the IEEE, Vol. 66, No. 11, November 1978, pp. 1386-1408

CLAR82 Clark, David D., "Names, Addresses, Ports, and Routes", in DDN Protocol Handbook, Vol. 3, December 1985, pp. 3-27 - 3-40

DALA81 Dalal, Yogen K. and Printis, Robert S., "48-bit Absolute Internet and Ethernet Host Numbers", Proceedings of the 7th Data Communications Conference, October 1981, pp. 240 - 245

GEE80 Gee, KCE, An Introduction to Open Systems Interconnection, The National Computing Centre Limited, 1980

GELO87 Gelotte, Michael J., A Strategy for Testing Electronic Mail Gateway Architectures, BYU Computer Science Department Master's Thesis, April 1987

GROE86 Groenbaek, Inge, "Conversion Between the TCP and ISO Transport Protocols as a Method of Achieving Interoperability Between Data Communications Systems", IEEE Journal on Selected Areas in Communications, VOL. SAC-4, NO. 2, March 1986, pp. 288-296

HIND83 Hinden, Robert, et al, "The DARPA Internet : Interconnecting Heterogeneous Computer Networks with Gateways", IEEE Computer, September 1983, pp. 38-48

HURS84 Hurst, Mark, "Sharing Data Between Microcomputers on a Local Net", Data Communications, September 1984, pp. 139-148

ISO84 International Organization for Standardization, International Standard 7498 : Information Processing Systems - Open System Interconnection- Basic Reference Model, available from American National Standards Association, Inc., 1984

ISO84B ISO Transport Protocol Specification, ISO DP 8073, RFC 905 available from SRI International, April 1984

ISO85 International Organization for Standardization, Draft International Standard 8348 : Information Processing Systems - Data Communications - Network Service Definition Addendum 2, available from American National Standards Association, Inc., 1985

ISO86 International Organization for Standardization, International Standard 8072 : Information Processing Systems - Open Systems Interconnection - Transport Service Definition, available from American National Standards Association, Inc., 1986

ISO86B Final Text of ISO DIS 8473, Protocol for Providing the Connectionless-mode Network Service, RFC 994 available from SRI International, March 1986

ISO87 International Organization for Standardization, Draft International Standard 8073 : Information Processing Systems - Open Systems Interconnection - Connection Oriented Transport Protocol Specification Addendum 2, available from American National Standards Association, Inc., 1987

KAHN72 Kahn, Robert E., "Resource-Sharing Computer Communications Networks", Proceedings of the IEEE, VOL. 60, NO. 11, November 1972, pp. 1397-1407

KNIG83 Knightson, Keith G., "The Transport Layer Standardization", Proceedings of the IEEE, VOL. 71, NO. 12, December 1983, pp. 1394-1396

MARI66 Marill, Thomas and Roberts, Lawrence G., "Toward a Cooperative Network of Time-Shared Computers", Proceedings - Fall Joint Computer Conference, 1966, pp. 425-431

MCCO87 McCoy, Wayne, Implementation Guide for the ISO Transport Protocol, RFC 1008 available from SRI International, June 1987

MIL83 "Military Standard Transmission Control Protocol", MIL-STD 1778, 26 October 1983, in DDN Protocol Handbook, Vol. 1, December 1985, pp. 1-147 - 1-324

MIL83B "Military Standard Internet Protocol", MIL-STD 1777, 12 August 1983, in DDN Protocol Handbook, Vol. 1, December 1985, pp. 1-67 - 1-146

MILL84 Mills, D.L., "Exterior Gateway Protocol Formal Specification", in DDN Protocol Handbook, Vol. 2, December 1095, pp. 2-495 - 2-524

OPPE81 Oppen, Derek C. and Dalal, Yogen K., The Clearinghouse : A Decentralized Agent for Locating Named Objects in a Distributed Environment, available from Xerox Office Products Division, October 1981

PISC86 Piscitello, David M., et al, "Internetworking in an OSI Environment", Data Communications, May 1986, pp. 118-136

POST80 Postel, Jonathan B., "Internetwork Protocol Approaches", IEEE Transactions on Communications, VOL. COM-28, NO. 4, April 1980, pp. 604-611

REDE83 Redell, David D. and White, James E., "Interconecting Electronic Mail Systems", IEEE Computer, September 1983, pp. 55-63

ROBE70 Roberts, Lawrence G. and Wessler, Barry D., "Computer Network Developement to Achieve Resource Sharing", Spring Joint Computer Conference, 1970, pp. 543-549

ROBE74 Roberts, Lawrence G., "Data by the Packet", IEEE Spectrum, February 1974, pp. 46-51

SCHN83 Schneidewind, Norman F., "Interconnecting Local Networks to Long-Distance Networks", IEEE Computer, September 1983, pp. 15-24

SHEL82 Sheltzer, Alan, et al, "Connecting Different Types of Networks with Gateways", Data Communications, August 1982, pp. 111-122

SHOC78 Shoch, John F., "Inter-Network Naming, Addressing, and Routing", The Proceedings of COMPCON, Fall 1978, pp. 389-395

SHOC79 Shoch, John F., "Packet Fragmentation in Inter-Network Protocols", in Computer Networks, Vol. 3, North-Holland Publishing Company, 1979, pp. 3-8

STAL84 Stallings, William, "A Primer : Understanding Transport Protocols", Data Communications, November 1984, pp. 201-215

STAL85 Stallings, William, Data and Computer Communications, Macmillan Publishing Co., 1985

STAL85B Stallings, William, "Can We Talk", Datamation, October 15, 1985, pp. 101-106

STAL87 Stallings, William, Local Networks : An Introduction, Macmillan Publishing Co., 1987

SUNS77 Sunshine, Carl A., "Interconnection of Computer Networks", in Computer Networks, Vol. 1, North-Holland Publishing Company, 1977, pp. 175-195

SUNS78 Sunshine, Carl A. and Dalal, Yogen K., "Connection Management in Transport Protocols", in Computer Networks, Vol. 2, North-Holland Publishing Company, 1978, pp. 454-473

SUNS80 Sunshine, Carl A., "Current Trends in Computer Network Interconnection", in Advances in Data Communications Management, Heyden & Sons, pp. 147-154, 1980

SUNS81 Sunshine, Carl A., "Transport Protcols for Computer Networks", in Protocols and Techniques for Data Communication Networks, F. Kuo-editor, Prentice-Hall, Inc., pp. 35-77, 1981

TANE81 Tanenbaum, Andrew S., Computer Networks, Prentice-Hall, Inc., 1981

WARE83 Ware, Christine, "The OSI Network Layer : Standards to Cope with the Real World", Proceedings of the IEEE, Vol. 71, No. 12, December 1983, pp. 1384-1387

WEIS87 Weissberger, Alan J. and Israel, Jay E., "What the New Internetworking Standards Provide", Data Communications, February 1987, pp. 141-156

WITT86 Witt, Michael, "Moving From DOD to OSI Protocols : A First Step", Computer Communication Review, April/May 1986, pp. 2-7

XERO81 Internet Transport Protocols, Xerox System Integration Standard 028112, Xerox Corporation, December 1981

A META-PROTOCOL ARCHITECTURE FOR CONNECTING COMPUTER NETWORKS

J. Brad Lindsey

Department of Computer Science

M.S. Degree, December 1987

ABSTRACT

Accompanying the proliferation of computer networks has been a movement to connect them into cooperating internets. However, when attempting to do so, the different protocols used to satisfy these once isolated networks are found to incompatible. Due to its reliable nature, the transport layer from ISO's OSI Reference Model is chosen as the point of attachment for subnets and internet gateways. In this role, it is expected to supply traditional transport and inherited services. A meta-protocol architecture is proposed to relay these services from one subnet to the next, until internet messages arrive at their destination. The architecture is based upon each subnet providing two conversion routines -- one from the subnet protocol to the meta-protocol, the other, back to its own protocol. A simulated internet, demonstrating the capabilities of the meta-protocol approach, is described.

Committee Approval:

Robert N. Linebarger

Robert N. Linebarger,
Committee Chairman

Evan L. Ivie

Evan L. Ivie, Committee Member

Gordon E. Stokes

Gordon E. Stokes,
Graduate Coordinator